Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6651137, 13 pages
https://doi.org/10.1155/2021/6651137

Research Article

Data Authentication for Wireless Sensor Networks with High Detection Efficiency Based on Reversible Watermarking

Guangyong Gao,[1,2,3] **Zhao Feng**,[1,3] **and Tingting Han**[1,3]

[1]*Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China*
[2]*School of Information Science & Technology, Jiujiang University, Jiujiang 332005, China*
[3]*Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing 210044, China*

Correspondence should be addressed to Guangyong Gao; gaoguangyong@163.com

Received 16 December 2020; Accepted 21 May 2021; Published 1 July 2021

Academic Editor: Alexandros G. Fragkiadakis

Data authentication is an important part of wireless sensor networks (WSNs). Aiming at the problems of high false positive rate and poor robustness in group verification of existing reversible watermarking schemes in WSNs, this paper proposes a scheme using reversible watermarking technology to achieve data integrity authentication with high detection efficiency (DAHDE). The core of DAHDE is dynamic grouping and double verification algorithm. Under the condition of satisfying the requirement of the group length, the synchronization point is used for dynamic grouping, and the double verification ensures that the grouping will not be confused. According to the closely related characteristics of adjacent data in WSNs, a new data item prediction method is designed based on the prediction-error expansion formula, and a flag check bit is added to the data with embedded watermarking during data transmission to ensure the stability of grouping, by which the fake synchronization point can be accurately identified. Moreover, the embedded data can be recovered accurately through the reversible algorithm of digital watermarking. Analysis and experimental results show that compared with the previously known schemes, the proposed scheme can avoid false positive rate, reduce computation cost, and own stronger grouping robustness.

1. Introduction

Compared with the increasingly rich functions and uses of current network [1, 2], the main function of wireless sensor networks (WSNs) is only to transmit real-time data. Its main task is to collect the measured data sensed by the nodes in the monitoring area to the sink node and then send it to the receiver through the Internet. Because of the obvious advantages of transmitting the real-time data, WSNs have been widely used in many industries, such as military, transportation, medical, national defense, and smart home, which greatly improves the work efficiency and the speed of social development. However, with the increasingly widespread application of WSNs, the security problems are being gradually exposed [3]. There will be great huge destruction once the real-time data is tampered with by hackers. Therefore, it is necessary to conduct integrity data authentication for WSNs, but the traditional cryptography technology is not suitable for this network because of its complex algorithm and high cost [4]. In recent years, information hiding technology which is a method to protect communication security has been paid great attention by researchers [5]. Digital watermarking technology is an important branch of information hiding technology. The purpose is to embed specific digital signals into digital products to protect the copyright or integrity of products. Its obvious advantage is the low cost of algorithm calculation and communication. In addition, digital watermarking technology has been applied in WSNs since the adjacent data in the sensor are closely related. If the traditional watermarking is used, the data will be changed slightly but irreversibly, which is unacceptable for some special applications such as war and medical treatment [6, 7]. Therefore, reversible digital watermarking technology is the most suitable choice for WSNs.

In [8], an information hiding method based on spread spectrum, which embeds watermarking into DC component

to achieve stronger robustness, is proposed. Although the scheme does not generate additional cost, the original data will be destructed after embedding the watermarking. And the receiver cannot recover the data, so the scheme is not suitable for applications with high precision requirements. Affected by this situation, reversible digital watermarking technology has become a new direction in the field of information security research which can recover the data completely without any extra cost. In the early stage, reversible digital watermarking based on lossless compression was proposed firstly. On this basis, Celik et al. proposed a new scheme of LSB [9]. By compressing the signal part which is easy to be affected by embedding distortion and transmitting it as part of the payload, lossless recovery is achieved, which improves the compression efficiency and increases the embedding capacity, but the robustness of the scheme is not strong. Then, a fragile chain watermarking scheme [10], which divides the data into several data streams, generates the watermarking by using hash function and realizes the integrity authentication by embedding the watermarking into a hash chain connected before and after.

Tian proposed a reversible digital watermarking based on difference expansion [11]. The algorithm divides the image pixels into two groups and then traverses each group of pixels (x, y) to embed secret information. The extraction processing is still traversing the pixel groups according to the embedding process, then extracting the secret information, and restoring the pixel group. The method is controllable and can be embedded into the watermarking easily. Alattar used reversible wavelet transform on the basis of Tian to further improve the embedding capacity [12]. Wang et al. proposed a reversible watermarking algorithm based on dynamic prediction-error expansion [13], which firstly estimates the pixel with the smallest watermarking distortion to ensure the minimum distortion as far as possible. Dragoi and Coltuc proposed an algorithm based on the difference expansion of local prediction method [14] which provides a local adaptive prediction method and selects the prediction-error extension within the threshold range to embed secret information, and the hiding information capacity is positively correlated with the threshold range.

Liu et al. proposed a reversible watermarking scheme for data authentication in wireless body area network [15]. In this scheme, the data are grouped according to the fixed size to improve the grouping efficiency, histogram shift technology is used to avoid possible underflow or overflow, local map is generated to recover the shifted data, and the generated watermarking and integrity authentication are embedded in chain mode. In [16], Wu et al. proposed an authentication algorithm based on cyclic redundancy check (CRC) and reversible watermarking to solve the problem of data integrity authentication in WSNs. In this algorithm, sensor node is responsible for data stream grouping and watermarking embedding, and sink node is responsible for authentication and recovery of received data group. In order to reduce the computational complexity of sensor nodes as much as possible, the watermark is generated by calculating CRC code of data group, and the embedding watermark method is also implemented according to reversible watermarking. Jiang et al. proposed a scheme that combines homomorphic elliptic curve encryption algorithm and reversible digital watermarking to verify the integrity of wireless sensor network data [17]. The watermarking is generated by chaotic sequence, and the segmented data are embedded, respectively. At the same time, the difference of the original data is encoded, and then, the encoded data is encrypted by ellipse. All the data are fused in the cluster head node and sent to the base station. The data is recovered by reverse operation in the base station. The algorithm of Shi and Shao [18] is a typical example of WSN data using reversible watermarking for verification. It dynamically groups the data through synchronization points. Two consecutive groups of data are combined into a verification group. The generator group is responsible for generating the watermarking sequence and then embedding the watermarking into the carrier group. When any group of data is tampered, the receiver can detect the tampering because the watermarking verification would not be successful. The algorithm can effectively verify the integrity of the data. However, due to the uncontrolled length of the dynamic groups and the large length difference between the groups, it is easy to introduce the imperfect embedding of watermarking information. Moreover, the carrier group data in the transmission process will be changed slightly after embedding the watermarking, which will lead to generate fake synchronization points easily and result in the confusion of groups and high false positive rate. In this paper, a novel scheme is proposed which is aimed at data integrity authentication with high detection efficiency (DAHDE) in WSNs. The DAHDE combines dynamic grouping technology with double verification algorithm and uses the flag check bit to deal with fake synchronization points.

The main contributions of this paper can be summarized as follows:

(1) The DAHDE improves the robustness of dynamic grouping verification and reduces the false negative rate

(2) Using the flag check bit, the DAHDE can avoid the false positive rate caused by the appearance of fake synchronization points

The rest of this paper is organized as follows. Section 2 describes the proposed algorithm. Experimental results and analysis are shown in Section 3, and Section 4 presents the conclusion and summarizes this paper.

2. Proposed Algorithm

The first subsection of this chapter introduces the basic principle of the prediction-error expansion in reversible watermarking algorithm, the second subsection describes the structure of the WSNs, and Sections 2.3–2.6 explain the specific steps of DAHDE.

2.1. Prediction-Error Expansion in Image Watermarking. The prediction-error expansion method used in this paper is derived from the information hiding technology of grayscale image. In order to hide the information in a grayscale image,

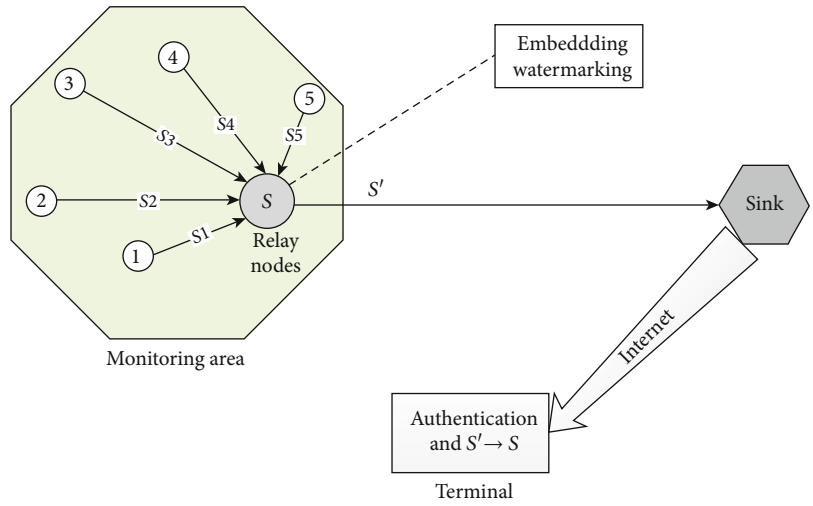

Figure 1: Wireless sensor network structure.

the prediction-error expansion technology firstly scans the image according to the specified order to get the pixel value and then embeds the watermarking into the mathematical difference between the two adjacent pixels. Set the pixel value as x and get its predicted value $\bar{x}$ through the prediction formula.

In Eq. (1), pe is the difference between the predicted value $\bar{x}$ and the actual data x.

$$pe = x - \bar{x}. \tag{1}$$

After pe is shifted one bit to the left according to the binary format, one bit watermarking w is embedded into its vacant LSB in Eq. (2).

$$pe' = 2 \times pe + w. \tag{2}$$

x' is the updated data calculated by

$$x' = pe' + \bar{x}. \tag{3}$$

So far, in the transmission process, the pixel value changes slightly to achieve information hiding. The receiver needs the following Eq. (4) to recover the data when decoding.

$$x = x' - \left\lfloor \frac{pe'}{2} \right\rfloor - w. \tag{4}$$

According to the above recovery method, when the transmission information is complete, the watermarking can be extracted correctly, and the information can be recovered completely by reversible formula. With this idea, and the adjacent data in WSNs have a high degree of correlation, the prediction-error expansion method can be used, too. The watermarking is generated from the data and embedded into the data. Once the sensory data stream is tampered, the receiver can detect it sensitively when decoding.

2.2. Wireless Sensor Network Structure. There are three important types of nodes in the WSNs, and these are sensor node, relay node, and sink node, respectively. The sensor nodes are distributed in a certain monitoring area, and the real-time data will be sent to the relay nodes by the sensor nodes in the form of data packets. The relay nodes will send the data stream to the sink nodes, and then, the sink nodes will fuse the data and transmit it to the terminal through network transmission. The distribution structure and data transmission processing of WSNs are shown in Figure 1. In addition, the reversible watermarking is embedded during the relay nodes, and finally, watermarking extraction and data restoration are carried out at the receiving end.

Among them, due to the particularly vulnerable characteristics of the sensor node, it will die after the end of the life cycle. The reason why the traditional encryption authentication method mentioned previously is improper for WSNs is that the resources of sensor node are limited and the computing ability is poor. Once a sensor node sends out the monitoring data, it will form a sensory data stream with the data of other nodes in the area. Our requirement for the security performance of WSNs is that the data stream should not change during transmission. When the data is lost or tampered by hackers, the receiver can quickly detect the tampered position and deal with it when receiving the data and verifying the data.

2.3. Grouping Scheme. Real-time sensory data stream is continuous and large, for example, when measuring temperature, each sensor node may get and transmit two or more data in one minute. Based on the idea of "watermark is born in the data and embedded into the data," when grouping the sensory data stream, a verification group is composed of two adjacent groups. The first group is the generator group, which is responsible for generating the watermarking sequence, and then, the watermarking sequence will be embedded into the second group that is named carrier group. Whether the watermarking is generated or embedded, each data in the sensory data stream is involved. Therefore, when data is tampered, the receiver can detect the tampering

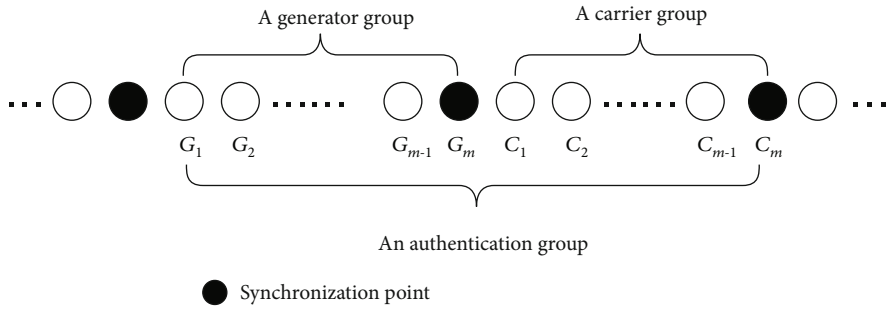

Figure 2: Example of an authentication group.

1. i⟵1
2. groupflag ⟵0
3. while (Stream S is not over) do
4. if (($m/2 < i <3m/2$ && md5(S_i) % m ==0 && groupflag ==0)||i ==$3m/2$)
5. spflag ⟵0//S_i is the synchronization point of generator group;
6. i ⟵0
7. groupflag ⟵1
8. else if (($m/2 < i <3m/2$ && md5(S_i) % m ==0 && groupflag ==1)||i ==$3m/2$)
9. spflag ⟵ 1//S_i is the synchronization point of carrier group;
10. i ⟵0
11. groupflag ⟵0
12. end if
13. i ⟵ i+1
14. end while

Algorithm 1: Localizing synchronization points

through decoding operation. In addition, in order to avoid the tracker to find the grouping mode, this design adopts dynamic grouping, and it determines the position of the synchronization point by judging the hash value of the data S_i. The DAHDE uses the MD5 function to calculate the hash value, and the length of the MD5 value of any form of data is 32 bits in hexadecimal and 128 bits in binary. Equation (5) is used to localize the synchronization point, which is shown as follows:

$$\text{MD5}(S_i)\%m == 0, \tag{5}$$

where MD5() is the function to calculate the hash value of the data, m is the group parameter, and % denotes modular division calculation. When the MD5 value of the sensory data stream S_i satisfies Eq. (5), the data will be regarded as the synchronization point. In this way, the sensory data stream can be dynamically grouped. Due to the random characteristics of MD5 value, the tracker cannot find out the value of the grouping parameter m. The dynamic grouping model is shown in Figure 2. As for the average length of the group, the probability that each data satisfies Eq. (5) is $1/m$, so the probability that each data is a synchronization point is $1/m$, so the average length of the group is m. The DAHDE sets the minimum and maximum group length as $m/2$ and $3m/2$, respectively, so as to ensure the robustness of grouping, avoid the trouble of large group length difference due to continuous special data, and effectively reduce the false negative rate, which will be explained in detail in Section 3. Pseudo code for localizing synchronization points is presented in Algorithm 1.

2.4. Flag Check Bit. If the operation is done only according to the algorithm above, the result of the operation will be inaccurate. The reason is the appearance of the fake synchronization point. When tampering data produces synchronization points which do not exist in the sensory data stream, it will cause temporary confusion of packets. Due to the existence of double verification, it will be judged as tampering, and the subsequent data can be checked normally. However, when the watermarking is embedded into group data, it is very likely that a fake synchronization point will appear. This fake synchronization point has nothing to do with tampering, only because of the algorithm. When encountering such fake synchronization points, it will be judged that the current verification group has been tampered. However, the actual data has not been tampered, which leads to a high false positive rate. Therefore, such fake synchronization points must be able to be recognized by the decoder. Because the data of WSNs have the characteristics of exposure, we cannot add identifier and other easily captured information in the data.

In order to solve the problem, an extra decimal place is added to the last bit of the sensory data stream as a flag check bit when transmitting data. The unique function of the flag check bit data is to judge the fake synchronization point. After the carrier group data is changed by embedding watermarking, if it satisfies Eq. (5), the data becomes a fake synchronization point, and a will be added to its flag check bit.

Input: original data stream S
Output: stream S'
while (Stream S is not over) do
1. if (spflag ==0)
2. $G \longleftarrow S_1 \cdots S_{i-1}$
3. $\bar{S} \longleftarrow 0$.
4. for $k \longleftarrow 1$ to $i-1$ do
5. $H_k \longleftarrow \text{hash}(G_k, d)$
6. $\bar{S} \longleftarrow (\bar{S} + G_k)\ /2$
7. end for
8. $H=H_1 \oplus H_2 \oplus \cdots H_m // H = \{W_r, 1 \le r \le d \ \&\ W_r = 0 || 1\}$;
9. else if (spflag ==*1*)
10. $C \longleftarrow S_1 S_2 \cdots S_{i-1}$
11. for $j \longleftarrow 1$ to $i-1$ do
12. $pe \longleftarrow C_j - \bar{S}$
13. $pe' \longleftarrow pe \times 2 + W_j$
14. $C_j' \longleftarrow \bar{S} + pe'$
15. $\bar{S} \longleftarrow (\bar{S} + C_j)\ /2$
16. if ($md5(C_j')$ % m ==*0*) //C_j' is a fake synchronization point;
17. $C_j' \longleftarrow \text{fcb}(a) + C_j'$ //Add fcb(a) to the flag check bit of C_j';
18. else
19. $b \longleftarrow$ rand() & $b \neq a$//b is a random number not equal to a;
20. $C_j' \longleftarrow \text{fcb}(b) + C_j'$
21. end if
22. $S_j' \longleftarrow C_j'$
23. end for
24. end if
25. end while

Algorithm 2: Encoding

The fixed number a that we set in advance is between 0 and 9. For example, data 24.14 changes to 24.56 because it was embedded watermarking, which is a fake synchronization point, then its state is 24.56a during transmission, so the original data with two decimal places becomes three decimal places during transmission. Note that 24.56a does not mean 25.56 multiplied by a, it is a single number. For other data, since the presence of flag check bit requires that all data be equally accurate, a random number b which is between 0 and 9 but not equal to a is added to the flag check bit of other data. When decoding, the flag check bit needs to be determined firstly, and then, the flag check bit will be eliminated directly if it is not a. If it is a, the data will not be determined as synchronization point, and then, the flag check bit will be eliminated, too. In this way, after restoring data, the flag check bit no longer exists. It only exists in the transmission process, guarantees the localization of synchronization point at a pretty low cost, and meanwhile improves the stability of the grouping. The pseudo code of the flag check bit is shown in steps 18-23 of Algorithm 2. The function fcb() is used to calculate the flag check bit according to the decimal places of original data. For example, when the data stream is two decimal places, the fcb(a) means multiplying a by 1e-03.

2.5. Watermarking Generation and Embedding. When the watermarking is generated by generator group, the same fixed length is taken from the MD5 values of all the data in the group for XOR operation; then, the watermarking sequence with length d is obtained, such as $w_1 w_2 \cdots w_d$. Note that the d value and the starting and ending positions selected here are also preset that just like the value of m, and only the sensor node and the receiver know about them, while the tracker cannot judge how the watermarking is generated and what is its rule. At the same time, the mean value of all the data in the generated group is taken as the initial predicted value $\bar{s}$ of the corresponding carrier group.

pe is obtained by the difference between the first data of the carrier group and $\bar{s}$, and then, pe is shifted to the left by one bit, and the watermarking w_1 is added to its LSB. At this time, the sum of pe' and $\bar{s}$ after embedding watermarking is the first data of the updated embedding group. In addition, $\bar{s}$ needs to be updated as the operation progresses. $\bar{s}$ is averaged with the current data to get a new $\bar{s}$, and this new setting is to make the predicted value closer to the real data. The detailed process in this section is shown in Algorithm 2.

2.6. Watermarking Extraction and Data Restoration. As a technology to verify the integrity of WSN data, after the data is tampered, whether the tampering occurs in the generator group or the carrier group, the inconsistency of the two groups of watermarking will be detected when decoding. When the current data is detected to be tampered with, it is not allowed to end the detection, and it is not allowed to find the incorrect group in the subsequent verification. Therefore, in order to ensure the smooth progress of the subsequent verification, double verification technology is added to the

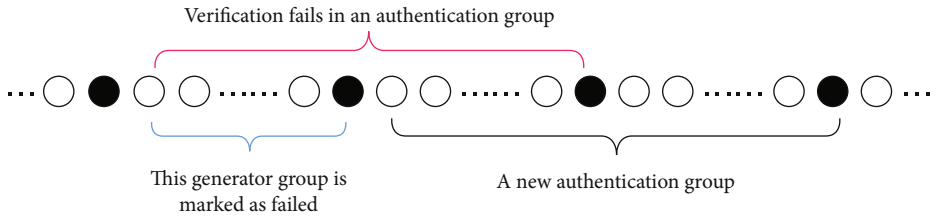

Figure 3: Double verification.

Input: stream S'
Output: recovered data stream S
1. while (stream S' is not over) do
2. if (spflag ==0)
3. do steps 3~10 in Algorithm 2;
4. else if (spflag ==1)
5. $C \longleftarrow S_1 \cdots S_{i-1}$
6. for $j \longleftarrow 1$ to $i-1$ **do**
7. $pe' \longleftarrow C_j - \bar{S}$
8. $w_j \longleftarrow \mathrm{LSB}(pe')$
9. if $(w_j \neq W_j)$
10. marking G as failed;
11. $G \longleftarrow C$//This failed carrier group becomes a new generator group;
12. break
13. else
14. $S_j \longleftarrow (pe' - w_j)/2 + \bar{S}$ //Restoring the original data;
15. end if
16. $\bar{S} \longleftarrow (\bar{S} + C_j)/2$
17. end for
18. end if
19. end while

Algorithm 3: Decoding

algorithm, as shown in Figure 3. When tampering is detected, the generator group in the authentication group will be judged as tampered at the first time, and then, the carrier group will be combined as a new generator group with the next group as a new verification group. Obviously, the watermarking verification of two groups of data with no operation association will fail. At this time, the current generator group which is the original carrier group, is judged to have been tampered, and then, a new verification is started which can also be grouped correctly. In other words, once any data in a certain verification group is tampered with, the whole verification group will be judged as not authenticated.

According to the generation and embedding of watermarking, the watermarking sequence generated by the generator group will be embedded into the carrier group in sequence. During the data transmission, the real value of the generator group remains unchanged, and the carrier group changes slightly due to the embedding of the watermarking. In the process of decoding, the watermarking and the initial prediction value $\bar{s}$ are obtained according to the same operation of generator group during encoding, and then, pe' is obtained by calculating the difference between C_i' and $\bar{s}$. At this time, the LSB of pe' is the corresponding 1-bit watermarking. As for data restoring, simply shift pe' to the right by one bit in binary and then add it and $\bar{s}$.

3. Experimental Results and Analysis

In order to find out the actual effectiveness of the algorithm for tampering detection, MATLAB is used to simulate the test. The data used are selected from the real data of Berkeley Laboratory, including temperature, humidity, light, and voltage. Before the experiment, the data need to be preprocessed, and the blank data items and the wrong format data items would be deleted. Only one of the data streams is shown in Figure 4 in order to show the experimental results more clearly. The .txt file is used to show the different states of data, which are original data, data during transmission, and restored data. This is only a screenshot of a partial verification group. After a long time of code testing, millions of level data can pass the experiment without any exception.

The analysis of experimental results includes the following parts: the robustness of dynamic grouping, the analysis of algorithm performance, and the advantages compared with the existing methods.

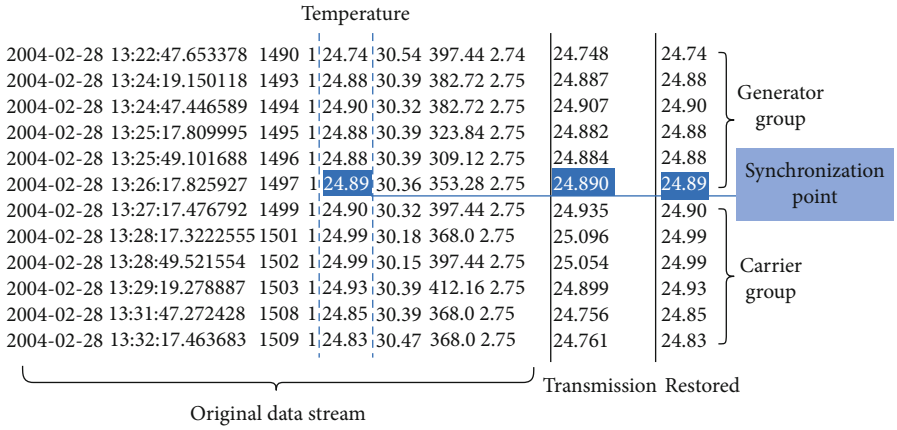

Figure 4: A part of verification group in simulation experiment.

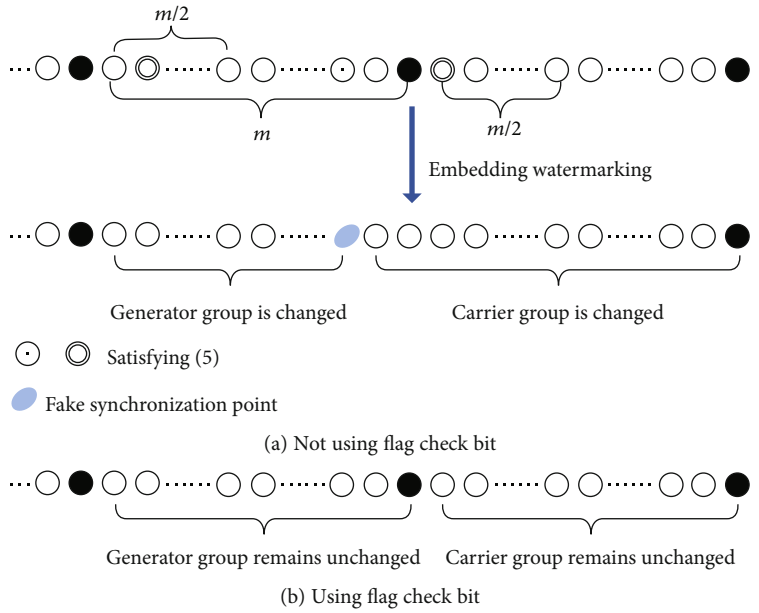

Figure 5: Robustness of grouping.

3.1. Robustness of Grouping. Dynamic grouping by calculating the hash value of sensory data stream is a relatively secret way. At the same time, in order to avoid too long or too short groups, a threshold is set when grouping, and the maximum and minimum values of group length l are limited. For sensory data stream, if a data is too short from the previous synchronization point ($l < m/2$), even if it satisfies Eq. (5), it will not be determined as a synchronization point. On the other hand, if the distance from the previous synchronization point is too long, there is no synchronization point until the $3m/2$ data. Whether the data satisfies Eq. (5), it is set as the synchronization point. Therefore, the minimum length is $m/2$ and the maximum length is $3m/2$, and the average value of group length can be calculated by

$$L = \sum_{l=m/2}^{3m/2} l \cdot \frac{1}{m} \cdot \left(1 - \frac{1}{m}\right)^{l} = m. \quad (6)$$

If the fake synchronization point appears too early or too late during watermarking embedding, it will not be judged as a synchronization point at the receiving end; that means the range of fake synchronization point can only appear is within $[m/2, 3m/2)$. Therefore, under the limit of the grouping threshold, the probability of fake synchronization points after watermarking embedding will be reduced. However, this reduction is insignificant compared to the flag check bit. The DAHDE uses the flag check bit that can avoid the fake synchronization point completely, and the sharp contrast effect with not using flag check bit is shown in Figure 5.

3.2. False Negative Rate. The three main ways that sensory data stream is tampered with during transmission are insertion, deletion, and modification. Insertion means inserting malicious forged data into the real data, which will greatly hinder the judgment of receiver for the data. Similarly, data

cannot be deleted or modified maliciously. Although the algorithm in this article can detect the above three tampering methods effectively, in most cases, the accuracy of an algorithm cannot reach 100%. The false negative rate may occur in occasional cases. When a piece of data is tampered with, the set of watermarking sequences may still remain unchanged, which will make the decoder unable to detect the tampering. Before starting to calculate the false negative rate of the three tampering methods, it is explained in advance that the most important influence on the false negative rate is the appearance of fake synchronization points. When encountering insertion tampering and modification tampering, the probability that the new data satisfies Eq. (5) is $1/m$, but is limited by the group length, and the probability of the data being judged as a synchronization point is only $1/(2m)$. Meanwhile, the DAHDE uses the flag check bit in transmission, the synchronization point is also affected by the last digit, and finally, the probability of the data being localized as the synchronization point is $9/(20m)$. On the contrary, the new data has the probability of $(20m-9)/(20m)$ is not the synchronization point.

3.2.1. Inserting a New Data Element. When the generator group is inserted into a piece of data that is not a fake synchronization point, the watermarking sequences of both the generator group and the carrier group will change, so the probability of fail of detection is $1/2^m$ which depends on the carrier group length. When the inserted data is a fake synchronization point, according to the double verification algorithm, the original verification group will be disrupted. Under normal circumstances, the new verification group can detect the watermarking sequence which does not match, and that means tampering has occurred. The probability of the verification succeeding will be affected by the position of the inserted data, which is determined by the length of the new carrier group. The P_{G1} in Eq. (7) represents the probability of fail of detection that caused by inserting data into the generator group that does not involve synchronization point, and the P_{G2} in Eq. (8) represents the probability of fail of detection that caused by inserting data into the generator group that involved synchronization point.

$$P_{G1} = \frac{20m\text{-}9}{20m} \times \frac{1}{2^m}. \tag{7}$$

$$\begin{aligned} P_{G2} &= \frac{9}{20m}\left(\frac{1}{2^{3m/2}} + \frac{1}{2^{3m/2-1}} + \cdots + \frac{1}{2^{m/2}}\right) \\ &= \frac{9}{20m}\sum_{i=0}^{m}\frac{1}{2^{m/2+i}} = \frac{9}{20m}\left(\frac{1}{2^{m/2-1}} - \frac{1}{2^{3m/2}}\right). \end{aligned} \tag{8}$$

When the carrier group is inserted into a data which is not a fake synchronization point, the insertion position is considered. More forward the position is, the shorter the detection time is, the lower the probability of fail of detection is. If the insertion data of the carrier group is determined as the synchronization point, the false verification group will pass the verification, and the false negative rate is high in this case. The P_{C1} in Eq. (9) represents the probability of fail of detection that caused by inserting data into the carrier group that does not involve synchronization point. The P_{C2} represents the probability of fail of detection that caused by inserting data into the carrier group that involved synchronization point. Obviously, the value of P_{C2} is $9/(20m)$.

$$\begin{aligned} P_{C1} &= \frac{20m-9}{20m}\cdot\left(\frac{1}{2^m} + \frac{1}{2^{m-1}} + \cdots + \frac{1}{2^{m/2}}\right) \\ &= \frac{20m-9}{20m}\cdot\sum_{i=0}^{m/2}\frac{1}{2^{m-i}} = \frac{20m-9}{20m}\left(\frac{1}{2^{m/2-1}} - \frac{1}{2^m}\right). \end{aligned} \tag{9}$$

Therefore, the P_{ins} is the average probability of fail of detecting insertion which can be calculated by

$$P_{\text{ins}} = \frac{1}{2}(P_{G1} + P_{G2}) + \frac{1}{2}(P_{C1} + P_{C2}) \approx \frac{9m+20}{40m^2}. \tag{10}$$

3.2.2. Deleting a Data Element. When a data element of the generator group which is not a synchronization point is deleted, it will not affect the grouping. Same as inserting and tampering, there will be a very low of false negative rate, and the probability is $1/2^m$. When the synchronization point of the generator group is deleted, the new generator group will be extended to $3m/2$. At this time, the false negative rate is determined by the length of the new carrier group $m/2$. The P_{G1} in Eq. (11) represents the probability of fail of detection that caused by deleting data from the generator group that does not involve synchronization point, and the P_{G2} in Eq. (12) represents the probability of fail of detection that caused by deleting data from the generator group that involved synchronization point.

$$P_{G1} = \frac{20m-9}{20m}\cdot\frac{1}{2^m}. \tag{11}$$

$$P_{G2} = \frac{9}{20m}\cdot\frac{1}{2^{m/2}}. \tag{12}$$

When a data element of the carrier group which is not a synchronization point is deleted, the false negative rate is related to the location of the deleted data. When the synchronization point of the carrier group is deleted, the new carrier group will be extended to the period of $3m/2$. At this time, the first $m-1$ data will be verified successfully, the probability of the later $m/2$ data will pass the verification is $1/2^{m/2}$. The P_{C1} in Eq. (13) represents the probability of fail of detection that caused by deleting data from the carrier group that does not involve synchronization point. The P_{C2} in Eq. (14) represents the probability of fail of detection that caused by deleting data from the carrier group that involved synchronization point.

$$P_{C1} = \frac{20m-9}{20m}\left(\frac{1}{2^{m-1}} + \frac{1}{2^{m-2}} + \cdots + \frac{1}{2}\right) = \frac{20m-9}{20m}\sum_{i=1}^{m-1}\frac{1}{2^i}. \tag{13}$$

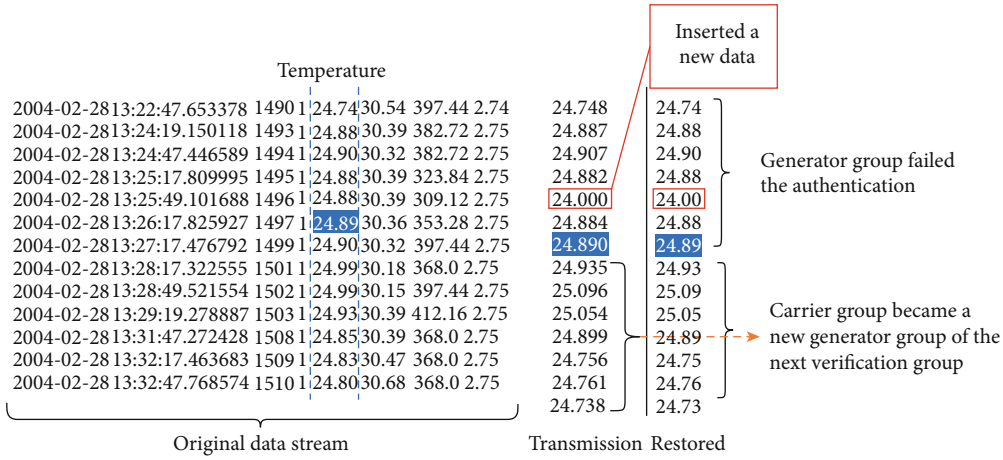

Figure 6: An example of insertion.

Table 1: False negative rate with different tampering amplitudes.

m	TA		
	(0, 1)	(1, 10)	Completely random
20	1.15%	1.10%	1.12%
30	0.78%	0.78%	0.75%
40	0.58%	0.55%	0.59%
50	0.46%	0.49%	0.46%
60	0.38%	0.39%	0.36%

$$P_{C2} = \frac{9}{20m} \cdot \frac{1}{2^{m/2}}. \tag{14}$$

Therefore, the P_{del} is the average probability of fail of detecting deletion which can be calculated by

$$P_{\text{del}} = \frac{1}{2}(P_{G1} + P_{G2}) + \frac{1}{2}(P_{C1} + P_{C2}) \approx \frac{1}{5m}. \tag{15}$$

3.2.3. Modifying a Data Element. Modifying a piece of data in the data stream is the most complicated situation. The main cumbersome point is the mutual modification of synchronization points and nonsynchronization points. Such modification is similar to deleting a piece of data and inserting a new piece of data. According to the calculation and the result of experiment, the average probability of fail of detecting modification is between insertion and deletion, about $9/(40m)$.

3.2.4. False Negative Rate in DAHDE. In most cases, tampering will be successfully detected, and an example of tampering is shown in Figure 6. One generator group was inserted a new data element during transmission; then, it would be marked as failed the authentication. The carrier group which composes a verification group with the generator group would become a new generator group because of the double verification. When false negative rate is generated, the verification group that is tampered will pass the authentication, so the false negative rate plays a crucial role of an algorithm. For the superior performance of DAHDE, the influence of the

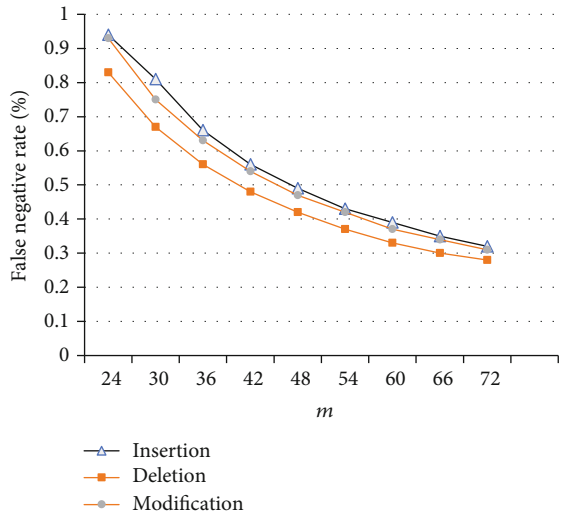

Figure 7: False negative rate with different m.

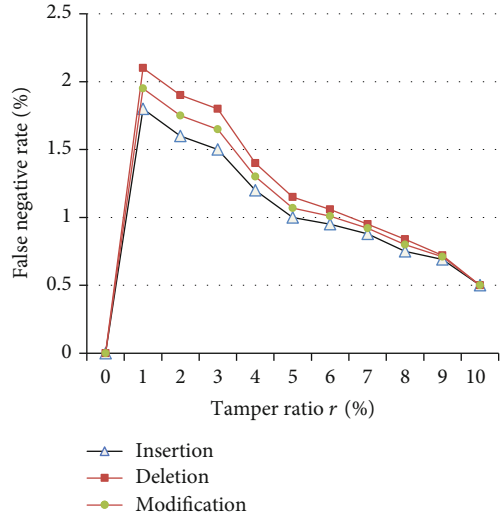

Figure 8: False negative rate with different r.

Table 2: Algorithm comparison with RWAS.

	Group length	Fake synchronization point	Watermarking
DAHDE	Setting the threshold, the group length is appropriate, grouping robustness is strong	Completely avoided	Recycling fixed length watermark
RWAS	Group length is not controllable	Unable to avoid	Sequentially embedding all generated watermarking

Table 3: Performance comparison with RWAS under different m conditions.

	False positive rate in DAHDE	False positive rate in RWAS	Average false negative rate in DAHDE	Average false negative rate in RWAS
$m = 20$	0	3.4%	1.2%	2.5%
$m = 30$	0	2.4%	1.1%	2.2%
$m = 40$	0	1.8%	0.6%	1.8%
$m = 50$	0	1.5%	0.5%	1.6%
Average rate	0	2.28%	0.85%	2.03%

grouping parameter m and the data tampering ratio r on the false negative rate of the algorithm is mainly considered. The following results are from a large number of fake tampering experiments. In the case of different m values, the relationship between false negative rate and tampering amplitude (TA) is shown in Table 1. It can be observed that even if tamper is small, DAHDE can still effectively detect it and will not bring about high false negative rate. Figure 7 shows the effect of the false negative rate on the size of the grouping parameter m, and Figure 8 shows the effect of the tampering data ratio on the false negative rate.

The length d of the watermarking sequence we selected in the experiment is 48 bits. Considering the efficiency of watermarking, the value range of grouping parameter m is within [24, 72). It can be seen from the experimental results that the false negative rate of the three tampering methods will not exceed 1%, and the false negative rate will decrease with the increase of m, but we cannot increase m indefinitely, because once detected for tampering, the entire verification group will be marked as uncertified, and excessive data will be wasted. Therefore, the final m value should be selected in accordance with the actual application requirements. The false negative rate will decrease as the proportion of tampered data increases, which means that this solution will not lose reliability due to the increase in attack frequency.

3.3. Overhead. An important advantage of DAHDE is that there is no extra transmission overhead. The group parameter m and watermarking sequence parameter d are preset and only known by the sender and receiver. In the transmission process, only the copy of the current verification group data is saved, and it will be cleared after the watermarking operation, so there will be no additional transmission overhead.

As for computational overhead, it will increase in the delay nodes because of the hash function. But the arithmetic operations included watermarking operations are considered as lightweight operations. In addition, compared with symmetric encryption and asymmetric encryption, the computational overhead of hash function is less in DAHDE.

3.4. Performance Comparison. By comparing the experimental results of DAHDE with the reversible watermarking authentication scheme in WSNs (RWAS) proposed in [18], it can be observed that the false negative rate and false positive rate are greatly reduced. The comparison of the two algorithms is shown in Table 2, and the performance comparison is shown in Table 3.

(1) *Robustness of Grouping.* Grouping scheme of RWAS is to use the synchronization point for dynamic groups; the mentioned threshold did not specify the scope that may lead to grouping is too long or too short. And it will affect the coding efficiency of the algorithm and data recovery rate. In DAHDE, based on the control threshold value and the relationship of m make a random group length in a reasonable range and effectively solve the problem of coding efficiency and cache. In addition, due to the limitation of the length of the verification group, the false negative rate has been reduced to some extent.

(2) *Processing the Appearance of Fake Synchronization Points.* In RWAS, if the hash value generated by the corresponding data S_i satisfies Eq. (5), S_i will be determined as the synchronization point. However, the original data may turn into fake synchronization point after embedding the watermarking, which will lead to the grouping disorder when extracting the watermarking. In the case that the watermarking is not attacked, some verification groups will be judged as tampering, and the original data cannot be completely recovered. In DAHDE, the data embedded with watermarking is processed during transmission, and a flag check bit is introduced, so that the receiver can skip the fake synchronization point

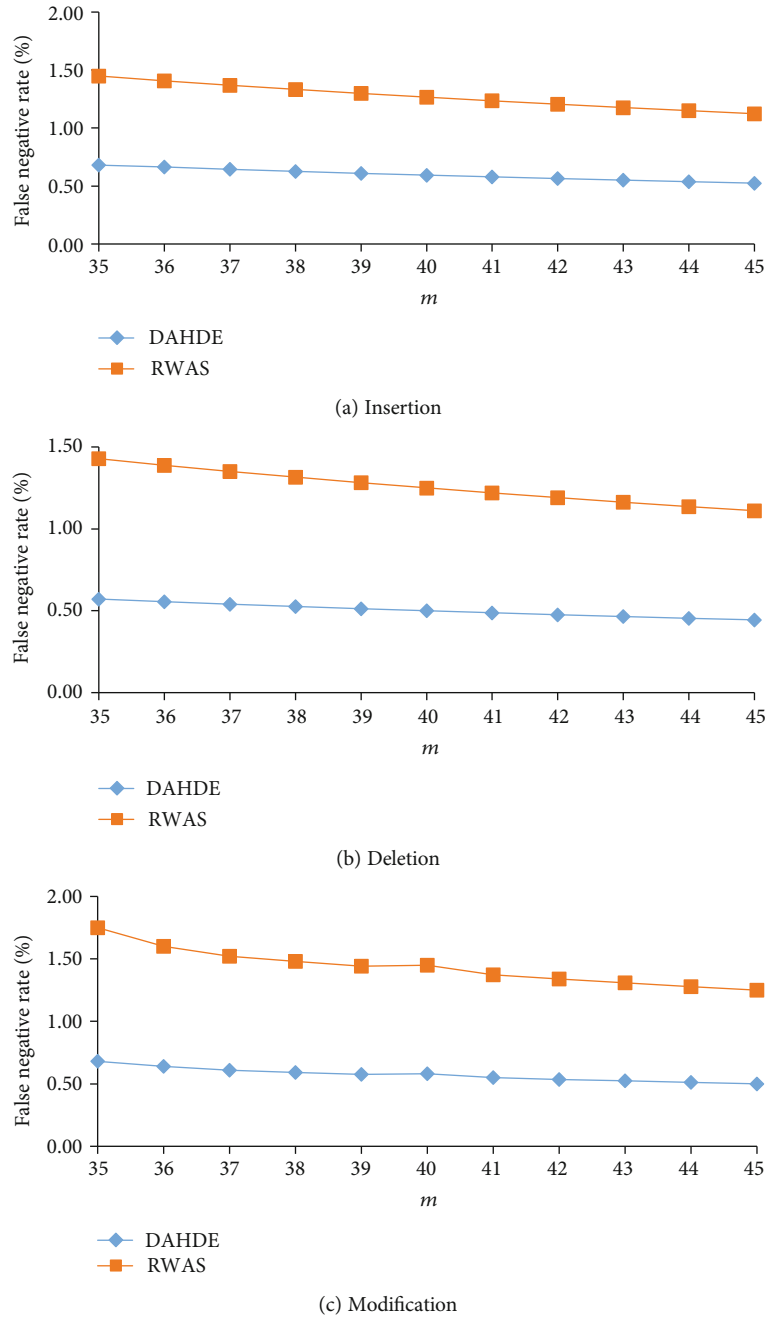

Figure 9: False negative rate under three kinds of tampering and comparison with RWAS.

when extracting the data. Without being tampered with, the data can be fully recovered, greatly reducing the failure rate of data authentication.

(3) *The Use of Watermarking*. In RWAS, all the generated watermarking (128 bits) are sequentially embedded without much consideration of the efficiency of the watermarking sequence. In DAHDE, the watermarking is embedded with the specified length of d, so that the size of d and m match. It can not only satisfy the efficiency of watermarking use, but also enhance the secrecy of watermarking sequence, so it can also help to improve the security of this algorithm.

The DAHDE solves the problem about fake synchronization point of RWAS, then the false positive rate is reduced to 0, and the false negative rate is only about 2/5 of that of RWAS. Table 3 shows the comparison of false positive rate and average false negative rate in three ways of tampering between DAHDE and RWAS when the parameter m is different. In addition, in order to prove the superiority of DAHDE in detail, we do three kinds of tampering experiments that the value range of grouping parameter m is within [35, 45] and

Table 4: Comparison with other schemes.

	DAHDE	RDWBP	LIAS	WSCRC	CWDM
Reversibility	Reversible	Reversible	Reversible	Reversible	Irreversible
Transmission overhead	0	0	Exist	Exist	0
Grouping	Dynamic	Static	Dynamic	Static	Dynamic

then compare them with RAWS. The result is shown in Figure 9.

Compared with DAHDE, Jiang et al.'s scheme [17] divides each data of the sensing node into slices (RDWBP), then embeds watermarking into two pieces of data divided by the data, and encrypts the data after embedding watermarking. For nodes, this will increase the calculation cost and improve the node's requirements for data cache capacity, which requires more sensor node resources. In addition, the scheme not only sends the encrypted and fused data to the base station, but also sends the unencrypted data only with watermarking, which will increase the extra overhead, consume the energy of the cluster head node, and reduce the transmission efficiency. In this paper, dynamic grouping is used in this algorithm. Jiang et al.'s algorithm is used for dynamic clustering of sensing nodes, and the sensory data is equivalent to static grouping, which has poor security and confidentiality.

In other several data authentication schemes [19], the comparison is mainly about reversibility, grouping, and overhead. Liu et al. proposed a Lightweight Integrity Authentication Scheme (LIAS) based on Reversible Watermark for Wireless Body Area Networks [15], and each group is composed of several packets. At the same time, each packet is marked with serial number, and the value of the flag DF is designed to mark whether the grouping is completed. Compared with DAHDE, the data packets to be cached are too large and will generate additional overhead in the scheme which is static grouping. Wu et al.'s scheme [16] is also static grouping, and the watermarking sequence is computed by CRC (WSCRC), which reduces the computational cost but increases the transmission cost. The scheme proposed by Guo et al. [10] is a classical irreversible watermarking authentication scheme (CWDM), which achieves dynamic grouping and has advantages in occupying sensor node resources. However, its irreversibility limits its application field. The specific comparison is shown in Table 4.

To sum up, DAHDE uses dynamic grouping to make it difficult for hackers to find out the rules of grouping and does not introduce transmission overhead. Furthermore, the adopted reversible watermarking technology ensures the complete restoration of data. Compared with the existing schemes, DAHDE solves the remaining problems, and the comprehensive performance (energy consumption, reversibility, security, etc.) is the best. Future work should focus on optimizing the grouping scheme, and even though hash function is much lighter than the traditional encryption method, there is still computation overhead that cannot be ignored. In addition, in the future work, we should break through the current limit of false negative rate to meet the security requirements of the algorithm to the utmost extent.

4. Conclusion

The data authentication scheme based on reversible digital watermarking in WSNs has passed the test successfully. It not only solves the problem of high false positive rate in the original group authentication scheme, but also ensures the integrity authentication of data with very low false negative rate. Flag check bit and double verification make DAHDE pretty robust, since the relay nodes transmit data packets immediately before the watermarking is embedded and merely cache a copy of the current data element, which will not affect the transmission speed. In addition, compared with the traditional encryption technology, the communication cost of digital watermarking is more lightweight, which fully satisfies the needs of WSNs.

Data Availability

The data used to support the findings of this study have not been made available because we need it for in-depth research, and the relevant experimental data in this paper is inconvenient to provide.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Jiangxi Key Natural Science Foundation under no. 20192ACBL20031, in part by the National Natural Science Foundation of China under Grant no. 61662039, in part by the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology (NUIST) under Grant no. 2019r070, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund.

References

[1] Z. Wu, G. Li, Q. Liu, G. Xu, and E. Chen, "Covering the sensitive subjects to protect personal privacy in personalized recommendation," *IEEE Transactions on Services Computing*, vol. 11, no. 3, pp. 493–506, 2018.

[2] Z. Wu, R. Wang, Q. Li et al., "A location privacy-preserving system based on query range cover-up or location-based services," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5244–5254, 2020.

[3] H. M. Xie, Z. Yan, Z. Yao, and M. Atiquzzaman, "Data collection for security measurement in wireless sensor networks: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2205–2224, 2019.

[4] C. Zhao, C. Wu, X. Wang et al., "Maximizing lifetime of a wireless sensor network via joint optimizing sink placement and sensor-to-sink routing," *Applied Mathematical Modelling*, vol. 49, pp. 319–337, 2017.

[5] S. Kim, R. Lussi, X. C. Qu, F. J. Huang, and H. J. Kim, "Reversible data hiding with automatic brightness preserving contrast enhancement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2271–2284, 2019.

[6] S. Yi and Y. C. Zhou, "Separable and reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 51–64, 2019.

[7] X. Li, J. Y. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Y. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Systems Journal*, vol. 14, no. 1, pp. 39–50, 2020.

[8] J. Huang, Y. Q. Shi, and Y. Shi, "Embedding image watermarks in dc components," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 974–979, 2000.

[9] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.

[10] H. P. Guo, Y. J. Li, and S. Jajodia, "Chaining watermarks for detecting malicious modifications to streaming data," *Information Sciences*, vol. 177, no. 1, pp. 281–298, 2007.

[11] T. Jun, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, 2003.

[12] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

[13] C. Wang, X. Li, and B. Yang, "Efficient reversible image watermarking by using dynamical prediction-error expansion," in *2010 IEEE International Conference on Image Processing*, pp. 3673–3676, Hong Kong, China, 2010.

[14] I. C. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1779–1790, 2014.

[15] X. Liu, Y. Ge, Y. S. Zhu, and D. Wu, "A lightweight integrity authentication scheme based on reversible watermark for wireless body area networks," *KSII Transactions on Internet and Information Systems*, vol. 8, pp. 4643–4660, 2014.

[16] H. Y. Wu, Y. Chen, and Z. M. Ji, "Wireless sensor networks authentication algorithm based on CRC and reversible digital watermarking," *Computer Applications and Software*, vol. 33, pp. 294–297, 2016.

[17] W. Jiang, Z. Zhang, and J. Wu, "Reversible digital watermarking-based protocol for data integrity in wireless sensor network," *Journal on Communications*, vol. 39, pp. 118–127, 2018.

[18] X. Shi and D. Xiao, "A reversible watermarking authentication scheme for wireless sensor networks," *Information Sciences*, vol. 240, pp. 173–183, 2013.

[19] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.