A novel watermarking scheme for authenticating individual data integrity of WSNs

Guangyong Gao[1,2*] and Min Wang[1]
[1] Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[2]School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, China
[e-mail: gaoguangyong@163.com]
[*]Corresponding author: Guangyong Gao

Received February 15, 2022; revised October 20, 2022; revised January 28, 2023; accepted February 16, 2023; published March 31, 2023

Abstract

The limited computing power of sensor nodes in wireless sensor networks (WSNs) and data tampering during wireless transmission are two important issues. In this paper, we propose a scheme for independent individual authentication of WSNs data based on digital watermarking technology. Digital watermarking suits well for WSNs, owing to its lower computational cost. The proposed scheme uses independent individual to generate a digital watermark and embeds the watermark in current data item. Moreover, a sink node extracts the watermark in single data and compares it with the generated watermark, thereby achieving integrity verification of data. Inherently, individual validation differs from the grouping-level validation, and avoids the lack of grouping robustness. The improved performance of individual integrity verification based on proposed scheme is validated through experimental analysis. Lastly, compared to other state-of-the-art schemes, our proposed scheme significantly reduces the false negative rate by an average of 5%, the false positive rate by an average of 80% of data verification, and increases the correct verification rate by 50% on average.

Keywords: Independent individual authentication, WSNs, digital watermark.

http://doi.org/10.3837/tiis.2023.03.014 **ISSN : 1976-7277**

1. Introduction

Currently, wireless sensor networks (WSNs) [1-5] have been widely used in the era of big data. Analyzing the various data collected in life can not only provide effective information but also help people improve their living standards. For example, when analyzing medical conditions [6], the data collected through the Internet can help doctors discover and resolve patients' diseases. In daily life, the smart home system [7] monitors the living conditions by analyzing the data collected by special sensors deployed in home appliances or living environments. In different applications, the security of the data stream is very important. Several methods have been proposed for protecting data.

For instance, Verma et al. [8] designed a unique type of encryption scheme, which implements privacy homomorphic encryption and end-to-end data protection. In work [9], the aggregate over multi-hop homomorphic encrypted data scheme is proposed. This scheme's principal goal is to ensure that each sensor data is aggregated within the WSN in a secure fashion, while fully preserving its privacy and integrity. Different from the traditional encryption fusion mechanism [10-13], the privacy scheme of wireless sensor network integrity verification based on homomorphic encryption has been studied. These protocols implement end-to-end security checks to ensure that data is not perceived during transmission. For wireless sensor networks, whether it is a conventional encryption algorithm or a data hiding security algorithm, communication capacity and computational overhead are inevitable. The application of cryptographic operations to computationally limited wireless sensors remains a challenge. Current research on low power devices [14] has done a lot of work including reducing energy consumption. It is well known that digital watermarking technology [15-16] is characterized by security, concealment and robustness. Some literature proposes simple and secure authentication schemes for data streams based on digital watermarking technology.

Digital watermarking based data integrity authentication of WSNs is mainly performed by data grouping. Likewise, Guo et al. [17] provided a chained group authentication scheme, which uses synchronization points to group data and achieves data integrity authentication. Hameed et al. [18] introduced a zero-watermark strategy, which is lightweight and robust against multiple types of data attacks compared with other schemes. Although grouping reduces the computational overhead of the sensor during the watermark calculation process, if one data in a group is abnormal, the data of the entire group will fail to be verified. In addition, if the flag of the group is tampered with, then at least two adjacent groups of data will be considered as tampered with. Therefore, data integrity authentication for wireless sensor networks requires a new solution to solve these problems.

The study of individual authentication of wireless sensor network data based on digital watermarking can solve the problems such as poor grouping robustness and high false positive during verification in grouping schemes. Recently, many attempts in literature have further studied the concept of individual authentication. For example, Zhang et al. [19] used a position random watermark to complete the data integrity authentication, which guarantees zero data interference at a reasonable cost. In addition, Hoang *et al.* [20] proposed a lightweight hybrid security scheme based on watermarking technology to protect perception data and defend nodes from cloning attacks. In the scheme [21], the signal is embedded into digital watermarking technique to uniquely represent the signal to identify the category of the vehicle. Chen *et al.* [22] proposed a WSN watermarking cryptosystem that provides lightweight data compression through zero-cost encryption. This scheme reduces the complexity of sensor node design and saves transmission power. Since no grouping is involved, the key advantage linked to individual integrity certification is free from the vulnerability of group tags to potential attacks. Even so, efficiency improvement for individual certification is a challenging task and serves as a bottleneck problem to be solved.

Data integrity authentication [23-25] is a network security requirement that cannot be ignored. Identity verification is a key technology for security measures. However, due to the limitations of memory, computing, and energy consumption, further research on the integrity of sensor data is particularly important. Hence, to overcome the current gap for the extra overhead of the grouping scheme and the high false positive rate of the individual verification scheme, a novel watermarking scheme for authenticating the individual data integrity of WSNs is proposed in this paper, where the integrity verification of each data item is carried out independently. The system model of the proposed algorithm is shown in **Fig. 1**, which is primarily composed of sensor nodes and a sink node. Initially, the Hash operation is conducted on a data item, and the obtained result is segmented. Next, each segment is folded into 1 bit, respectively, and the generated bit string is divided into two parts. Subsequently, two decimal numbers are calculated. The utilized embedding method replaces the lowest fractional parts of the data item with two generated decimal numbers, respectively. Finally, the order of fractional parts is randomly scrambled.

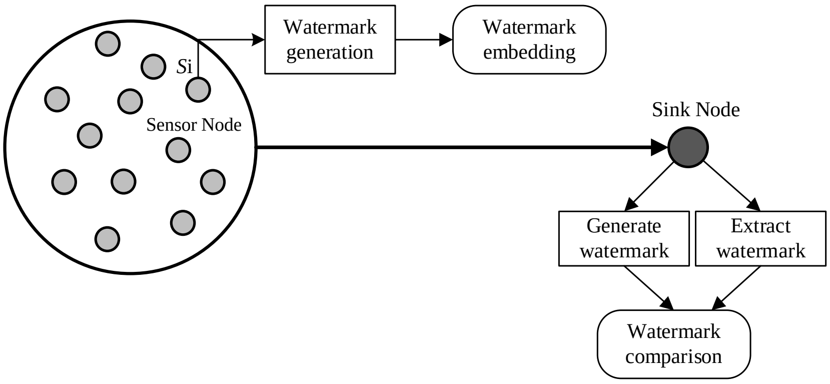


Fig. 1. Diagram of proposed scheme.

Experimental results show that the proposed scheme is better than previous work in terms of data integrity authentication. The proposed method can be well applied to the wireless sensor network, and the data obtained by the user after transmission through the network can be real and effective. In order to retain more useful data, we perform individual authentication on the data to be transmitted.

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

(1) The proposed method avoids the correlation between data, which makes the false positive rate equal to zero and greatly reduces the false negative rate in the verification process. Therefore, the ability to identify tampered data has been significantly improved.

(2) A novel watermarking scheme for authenticating individual data integrity of WSNs is proposed. In addition, compared with the grouping scheme, the proposed scheme improves the robustness of data integrity verification and reduces the computational overhead and storage capacity.

The rest of the paper is organized as follows: Section 2 thoroughly illustrates the proposed algorithm and theory analysis. In Section 3, the experimental results are presented and compared with other reported schemes. Lastly, the work is concluded in Section 4.

2. Related work

In this section, a comprehensive review to the previous related watermarking schemes in WSN is given. At the same time, the related work of different attacks on watermarking schemes in WSN is introduced.

2.1Attacks of WSNs

When sensors are exposed to external areas, they are often subject to malicious attacks. At the WSN network layer, the main attack methods are discarding, tampering and selective forwarding. Data tampering includes modifying data, deleting data, and adding data. Selective forwarding attacks result in some data packets not being sent to the base station, which destroys the integrity of the data. When data is intentionally modified during transmission to the receiving node, the results of data analysis can deviate from authenticity. When important data is deleted, it also affects the analysis of the problem. In addition, if data is deliberately inserted into the data stream, the extra data affects the analysis at the receiving end. Data integrity certification ensures that no changes have been made during data transfer. By verifying the watermark information, the authenticity of the data packet can be easily found. During group authentication, if the data of a certain group is tampered with, the data of this group will be discarded. For individual authentication, when a data item is tampered with, only the current data is discarded.

2.2 Previous WSNs Watermarking Schemes

Jiang et al. [26] processed the data stream through chaotic sequences and introduced homomorphic encryption to prevent valid data from being intercepted. Shi [27] generated watermarks by randomly selecting the data in queue. Due to existence of correlation among the data, the reported scheme greatly reduced false negative r ate but elevated the false positive rate, simultaneously. To improve further, Xiao et al. [28] first divided the data stream into arrays of length N and converts the arrays into character arrays. Then they multiplied the N×N matrix B and the groupings to select the data items for computing the watermark. Next, the hexadecimal number generated by the hash function is converted into a binary number. 128 bits are folded into 1 bit number by XOR operation for a total of n bits. Finally, n bits are embedded into n data items in the packet, respectively. If the watermark value is 1, a space is added after the data character, and if the watermark is 0, it is not added. When extracting a watermark, the watermark is extracted according to whether there is a space at the end of the character data. Xiao et al. [28] refined Shi's [27] method and reduced the overall error rate of data tampering detection. Although Xiao [28] embeds watermark on individual data to verify data integrity and improve the performance of the scheme, generating watermark by grouping increases memory consumption. In scheme Shi [29], sensor nodes first group data, and two adjacent data groups form a non-overlapping authentication group. The watermark bits are calculated from the first data group through the hash function, and the watermark is embedded in the second data group by the prediction error expansion technique. On the other hand, the receiver verifies the calculated and extracted watermark bits after synchronizing the data group, and finally restores the original data. However, if the group flag is corrupted then both groups of data will be authenticated incorrectly. Next, we will detail the specific steps of two related work, which are grouping scheme and individual work namely Jiang et al.'s method [26] and Shi's method [27], respectively.

2.2.1 Jiang et al.’ method [26]

A. Watermark generating and embedding

All nodes are clustered according to LEACH protocol, then the sensor data *N* is divided into n_1 and n_2 at the sensor node. Next, the watermark *w* is embedded into n_1 and n_2 using differential expansion technology:

$$h = n_1 - n_2 \tag{1}$$

$$h' = 2h + w \tag{2}$$

At the same time, the original data is homomorphically encrypted based on the elliptic curve at the sensor node. If n_1, $n_2 \in Q$ and $k \in K$, *K* is the key space. Then the additive homomorphic operation and multiplication homomorphic operation are respectively:

$$EK(n_1 + n_2) = EK(n_1) \oplus EK(n_2) \tag{3}$$

$$EK(n_1 \times n_2) = EK(n_1) \otimes EK(n_2) \tag{4}$$

where $\oplus, \otimes$ represent some kind of operation. Encryption technology can provide end-to-end data privacy protection.

Then, the watermark data and encrypted data are sent to the cluster head. The cluster head merges the received encrypted data to obtain aggregated ciphertext *C*, then arranges the received watermark data in a matrix manner. Finally, the aggregated ciphertext and watermark data are sent to the base station.

B. Data integrity authentication

The base station first decrypts the encrypted data, and then restores it to the original data through reverse mapping. In addition, the mapping table of cluster ID and node ID is searched in the base station to obtain the embedded watermark data. Next, the data is fragmented into n_1' and n_2', and the difference h' of the fragmented data is calculated. Finally, the node ID is determined and extracts the watermark data w':

$$w' = h' \bmod 2 \tag{5}$$

The base station calculates the chaotic sequence to obtain the embedded watermark *W* and compares it with the extracted w' to verify the integrity of the data.

2.2.2 Shi et al.’ method [27]

A. Watermark generating and embedding

For the sender's data element d_i, its copy c_i is first added to the queue buffer of length *N*. Then data is selected from the queue with probability *p*, which is used as the hash candidate set *C*. Next, calculate the hash value *H* of *C* and convert it to binary. *H* is folded into one watermark bit *w* by XOR operation:

$$w = b_1 \oplus b_2 \oplus \cdots \oplus b_i \tag{6}$$

Finally, each data element carrying watermark information is transmitted to the receiving end.

B. Watermark extraction and comparison

When the receiving end processes watermarked data d', it is first stored in the queue as the sending end. At the same time, the length of the queue is the same as that of the sender. Next, the candidate data is selected from the queue to participate in the hash calculation. H' calculated by Hash is also folded into one watermark bit w_1. Then, w_2 is taken from the LSB of the data and compared with the calculated w_1. If the calculated w_1 is equal to the extracted w_2, the data has not been tampered with during transmission.

Next, we briefly compare the main techniques of several previous WSN-based watermarking schemes. **Table 1** lists several main watermark-related operations in different schemes.

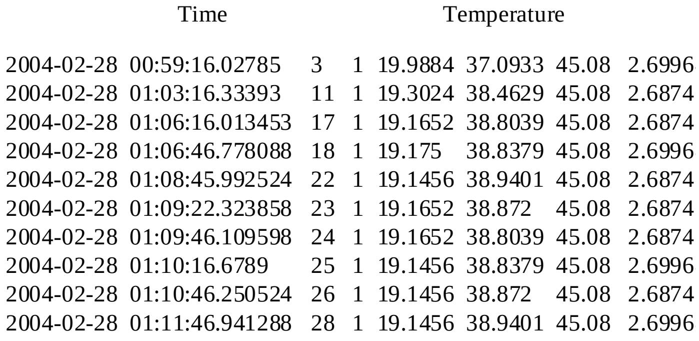
Table 1. Comparison of existing watermarking schemes

Schemes	Guo [17]	Jiang [26]	Shi [27]	Xiao [28]	Shi [29]
Watermark generated method	hash	chaotic sequence	hash	hash	hash
Data processing method	group	group	queue	group	group
Watermark embedded method	LSB	difference expansion	LSB	adding whitespace	difference expansion

As can be observed from **Table 1**, most of the watermarking schemes use group to process data, in which the grouping flag is easily corrupted and the grouping consumes memory, which is not suitable for low consumption WSNs. To solve the problems and deficiencies in the above literature, we propose the integrity verification of individual data items based on watermarking.

3. Proposed algorithm

This section describes the details of generating digital watermarking from data. Each sensor data not only generate corresponding watermark information, but also serve as a carrier for embedding watermarking. Therefore, the information embedding method must satisfy security and small volume capacity. Taking into account the comparison of watermarks in the scheme, we have made a new design for the position of the embedded watermark in the data. The watermark at the receiving end does not involve data calculation, it is placed in the meaningless decimal place of the sensor data and then randomly scrambled. **Fig. 2** shows the partial actual sensor data from Intel Berkeley Research Labs [30].



```
                  Time                           Temperature

2004-02-28  00:59:16.02785    3    1  19.9884  37.0933  45.08   2.6996
2004-02-28  01:03:16.33393    11   1  19.3024  38.4629  45.08   2.6874
2004-02-28  01:06:16.013453   17   1  19.1652  38.8039  45.08   2.6874
2004-02-28  01:06:46.778088   18   1  19.175   38.8379  45.08   2.6996
2004-02-28  01:08:45.992524   22   1  19.1456  38.9401  45.08   2.6874
2004-02-28  01:09:22.323858   23   1  19.1652  38.872   45.08   2.6874
2004-02-28  01:09:46.109598   24   1  19.1652  38.8039  45.08   2.6874
2004-02-28  01:10:16.6789     25   1  19.1456  38.8379  45.08   2.6996
2004-02-28  01:10:46.250524   26   1  19.1456  38.872   45.08   2.6874
2004-02-28  01:11:46.941288   28   1  19.1456  38.9401  45.08   2.6996
```

Fig. 2. Partial sensor data records of the experimental simulation.

3.1 Watermark generation and embedding

As depicted in **Fig. 3**, the sensor nodes collect N data records, which are denoted as S_i, S_{i+1}, ..., S_{i+N-1}. The array caches one record at a time. Each data record S_i collected by the sensor node includes the actual sensor data item d_i and time t_i, which arrives at the sink node through an insecure channel transmission.

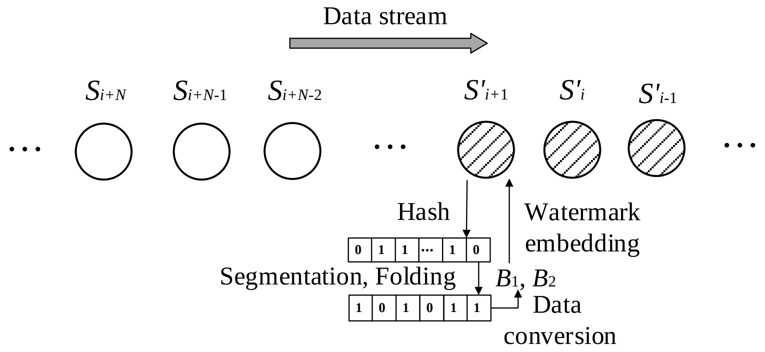


Fig. 3. Watermark generation and embedding.

Table 2. The system functions

Function	Description
Hash(x)	The x is used to generate a 128-bits binary number.
Slice (x)	The binary number x can be divided into six bit strings.
Folding (x)	0 or 1 bit can be generated by XOR operation among each bit from the bit string x.
Replace(a,b,c,d,x)	The third and fourth decimals c and d of x are replaced by a and b, respectively.
Merge(x,y)	Merge x and y into binary string.
Dec (x)	Convert binary bit string x into a decimal number.
RanScram(x)	The fractional parts of x are scrambled randomly.
RanScramInv(x)	The fractional parts of x can be scrambled inversely.

All relevant functions are listed in **Table 2**. First, we multiply sensor data item d_i by 100, which is then rounded down to an integer m. Later on, d_i is restricted to four decimals, either by zero padding or deleting fractional parts, where the third and fourth decimals are denoted as l_1 and l_2, respectively. In general, only first two fractional parts of d_i are considered meaningful. The watermark generation algorithm generates a 128-bits binary number M by Hash operation and m as shown in Eq. (7), and divides averagely M into six segments by Slice function (Eq. (8)). Correspondingly, the obtained six segments are folded to form six bits b_1, b_2, ..., b_6, as expressed in Eq. (9).

For example, considering d_i to be 19.7336, then m is 1973, and l_1, l_2 are 3 and 6, respectively.

$$M = \mathrm{Hash}(m) \tag{7}$$

$$X_i=\mathrm{Slice}(M)(i=1,2\cdots6) \tag{8}$$

$$b_i=\mathrm{Folding}(X_i)(i=1,2\cdots6) \tag{9}$$

Finally, the six bits are divided into two equal segments (three bits each). The first three bits (b_1, b_2, b_3) and last three bits (b_4, b_5, b_6) are converted into decimal numbers B_1 and B_2, respectively, which are noted as watermark, as given in Eq. (10).

$$B_k = \text{Dec}[b(k^2 : 3k)](k = 1, 2) \tag{10}$$

The watermark embedding is implemented by replacing the lowest two fractional parts i.e., l_1 and l_2 are replaced with B_1 and B_2, respectively (Eq. (11)). Eventually, after embedding, the order of fractional parts of a data item is randomly scrambled as expressed in Eq. (12). The function of the RanScram(x) that the order of bit string x is randomly changed. Therefore, the position of watermark is changed randomly to prevent the watermark from being identified and attacked, intentionally. Moreover, the final generated data item is indicated as d'_{ir} .

$$d'_i = \text{Replace}(B_1, B_2, l_1, l_2, d_i) \tag{11}$$

$$d'_{ir} = \text{RanScram}(d'_i) \tag{12}$$

Since six bits are generated by folding during the watermark generation process, two decimal numbers (B_1 and B_2) generated by the segment are in a range $0 \leqslant B_1$, $B_2 \leqslant 7$. Furthermore, the detailed process of watermark generation and embedding is described in **Algorithm 1**.

Algorithm 1: Watermark generation and embedding

Input: data item d_i; data stream length N; decimals l_1, l_2.

Output: embedded watermarked **packet** d'_{ir} .

```
for $i \leftarrow 1$ to $N$ do
    $m \leftarrow$ floor ($d_i \times 100$)
    $M \leftarrow$ Hash ($m$)
    $X_j \leftarrow$ Slice($M$) ($j$ =1, 2…,6)
    $b \leftarrow$ null
    for $j \leftarrow 1$ to 6 do
        $b \leftarrow$ Merge ($b$, Folding ($X_j$))
    end for
    $B_1 \leftarrow$ Dec ($b$ (1: $length(b)/2$))
    $B_2 \leftarrow$ Dec ($b$ ($length(b)/2+1$: end))
    $d'_i \leftarrow$ Replace ($B_1$, $B_2$, $l_1$, $l_2$, $d_i$)
    $d'_{ir} \leftarrow$ RanScram ( $d'_i$ )
end for
```

3.2 Watermark extraction and comparison

When the sink node receives data items d''_{ir} , it first restores fractional parts of sensor data item to the original order by an inverse scrambling as shown Eq. (13). Progressively, a watermark is generated (using an identical method employed in sensor node watermark generation) and compared with extracted watermark. The process for extracting and comparing the watermark for data stream can be viewed in **Fig. 4**.

$$d_i'' = \text{RanScramInv}(d_{ir}'') \tag{13}$$

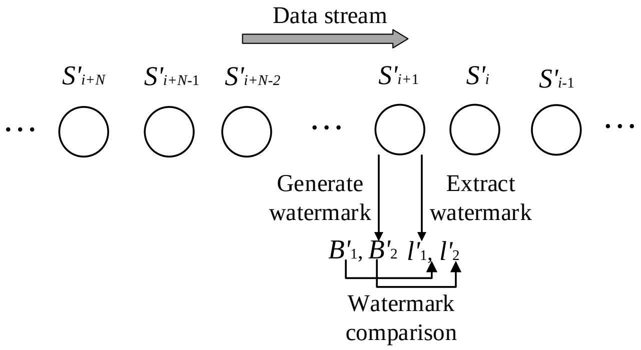


Fig. 4. Watermark extraction and comparison.

Next, a detailed description is presented for this process. The currently collected data item is computed in a same way as described for sensor node, and two decimal numbers are obtained denoted as B_1' and B_2'. Simultaneously, the lowest two fractional parts of this data item are extracted, labelled as l_1' and l_2'. Afterwards, the generated and extracted watermarks are compared to check for any possible malicious tampering.

If non-watermarked part of the data item is tampered during transmission, the generated B_1' and B_2' will be different from original watermark. Similarly, tampering of watermarked part of data item results in a variation of extracted watermark information l_1' and l_2', compared with the original content. Hence, proposed scheme detects the data tampering in two ways described above. The comprehensive process of watermark extraction and comparison for the given data stream is described in **Algorithm 2**.

Algorithm 2: Watermark extraction and comparison

Input: data item d_{ir}''; data stream length N.

Output: watermark comparison result *res*.

```
$d_i''$ ←RanScramInv ( $d_{ir}''$ )
for $i$← 1 to $N$ do
  $m$←floor ( $d_i''$ ×100)
  $M$←Hash ($m$)
  $X_j$ ←Slice ($M$) ($j$ =1, 2…,6)
  $b'$←null
  for $j$←1 to 6 do
    $b'$←Merge ($b'$, Folding ($X_j$))
  end for
  $B_1'$ ←Dec ($b'$ (1: $length(b')/2$))
  $B_2'$ ←Dec ($b'(length(b')/2+1$: end))
  $l_1'$ ←mod (floor ( $d_i''$ ×1000), 10)
  $l_2'$ ←mod (floor ( $d_i''$ ×10000), 10))
  if $B_1'$ == $l_1'$ && $B_2'$ == $l_2'$ then
    $res$←0
  else
```

```
    $res \leftarrow 1$
  end if
end for
```

3.3 False negative rate analysis

False negative occurs when a tampered data item is not recognized as a tampered one at the sink node. Although the Hash value calculated for tampered data items changes, bit flipping keeps the generated watermark value unchanged. Folding operation causes the generated bits of watermark information to shift from 0 to 1 or vice versa, with a probability of 1/2 for each bit shifting. It is not difficult to know that the number of bits flipped *K* follows a binomial distribution, where the probability distribution of '*K*' can be calculated using Eq. (14).

$$P(K=k)=\binom{M}{k}\left(\frac{1}{2}\right)^{M}\ (k=0,1,2\cdots M) \tag{14}$$

where *M* is the number of all watermark bits.

There exist three cases for data modification. The first case corresponds to modifying the data used to generate watermark only. In such case, computed six bits from the tampered data are flipped in a Folding process, and the decimal numbers generated by six bits are still equal to the relevant numbers of extracted watermark. The false negative rate (*FNR*) for this case is shown in Eq. (15). Likewise, the second case involves the modification of watermark part only. Since the watermark information is a decimal digit, ranging from 0 to 7, a changed watermark information will not be equal to the generated watermark, so *FNR* is 0 in this case. Ultimately, last case refers to modifying both data used to generate watermark and watermark part. Since the data of both parts are modified, *FNR* in this case is lower than the first case.

$$FNR=1-\sum_{k=1}^{M}P(k) \tag{15}$$

Insertion operation is similar to the last case of data modification. Since the inserted data items are not embedded with watermark information (i.e., extracting the watermark may be equivalent to computing the watermark), the *FNR* in this case is lower than that of data modification. Moreover, when a data item is deleted, it does not affect the authentication of other data items. Thus, deletion operation cannot generate false negative for the other data items, and *FNR* is 0.

3.4 False positive rate analysis

A false positive appears when a non-tampered data item is detected as a tampered item during extraction and comparison. For a data item d_i that is not tampered with, generated watermark is only related to the item itself, therefore, an extracted watermark from data item d_i will be equal to the generated watermark, i.e., a correct data item cannot be misjudged as a tampered item.

In terms of the analysis given above, all three tamper operations only influence the value of current data item. False positive cannot occur if the data items are not associated. Similarly, the false positive rate (*FPR*) is expressed in Eq. (16). According to above analysis, we conclude that *FPR* of the proposed scheme is 0.

$$FPR = \frac{n}{Truenum} \tag{16}$$

where n is the number of all non-tampered data items misjudged as tampered items and *Truenum* is the number of all true non-tampered items.

3.5 Correct verification rate analysis

The correct verification rate (*CVR*) is an indicator, which represents the ratio of the number of correctly verified data items to all test data items. To further explain, for different cases of tampering, the data verified by *CVR* is the total number of data received by the receiving end. The calculation of *CVR* is shown in Eq. (17). This indicator can measure the performance of the proposed solution to verify the integrity of each data item.

In addition, different from the two evaluation indicators mentioned above, *CVR* is also an evaluation to verify the integrity of all data. This indicator not only considers the case of false negative, but also involves the case of data being misjudged. Therefore, in the three cases of data tampering in the proposed scheme, the total number of correctly verified data is the total amount of data without false negatives.

$$CVR = \frac{v}{Allnum} \tag{17}$$

where v is the number of correct verified data items and *Allnum* is a number of all test data items. Algorithm 3 is presented to explain the details for the proposed calculation of evaluation indicators.

Algorithm 3: Calculation of evaluation indicators

Input: comparison results *res*; data stream length *N*.

Output: *FNR*; *FPR*; *CVR*.

```
for $i \leftarrow 1$ to $N$ do
  if $res(i) == 0$ then
    $ischange \leftarrow 0$
    // ischange is intermediate record variable
    for $j \leftarrow 1$ to length($numorder$) do // numorder is the index of all tampered data items
      if $i == j$ then
        $ischange \leftarrow 1$
        break
      end if
    end for
    if $ischange == 1$ then
      $fn\_num \leftarrow fn\_num + 1$ // fn_num is the number of false negative records
    else
      $fp\_num \leftarrow fp\_num + 1$ // fp_num is the number of false positive records
    end if
end for
$CVR \leftarrow (N\text{-}fn\_num - fp\_num) / N$
$FNR \leftarrow fn\_num / td\_num$
$FPR \leftarrow fp\_num / (N\text{-} td\_num)$ // $td\_num$ is the number of all tampered data items
```

4. Experiment results

In our experiments, the tampering attacks are divided into three categories i.e., modification, insertion, and deletion of data. The performance of the proposed scheme is tested by means of MATLAB simulations. The raw data stream used in experiments is from a real wireless sensor network deployed at Intel Berkeley Research Labs [30]. The experimental data is composed of 10,000 records collected by wireless sensors, where each data record includes some data items e.g., air temperature, voltage humidity, and time. For the sake of description, experimental results presented in this paper are only for one type of data item (temperature).

The performance of the proposed independent individual integrity authentication scheme is evaluated through tampering experiments and comparisons with other individual schemes. Experimental framework tests the capability of each scheme for data integrity authentication by relevant values of *FNR*, *FPR* and *CVR*.

4.1 Tampering detection

The independent variable for this experiment is tampering rate, which indicates the rate of the total amount of tampered data item divided by the total amount of data items. Additionally, the tampering rate is set from 10% to 50%. **Fig. 5** (a-c) demonstrates the experimental results for the three cases of modification attacks, respectively.

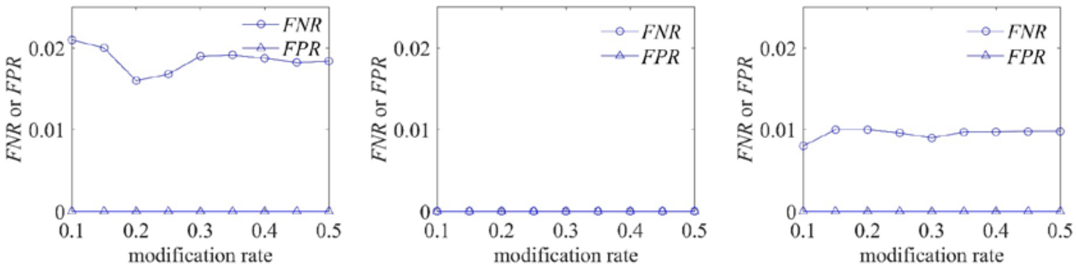
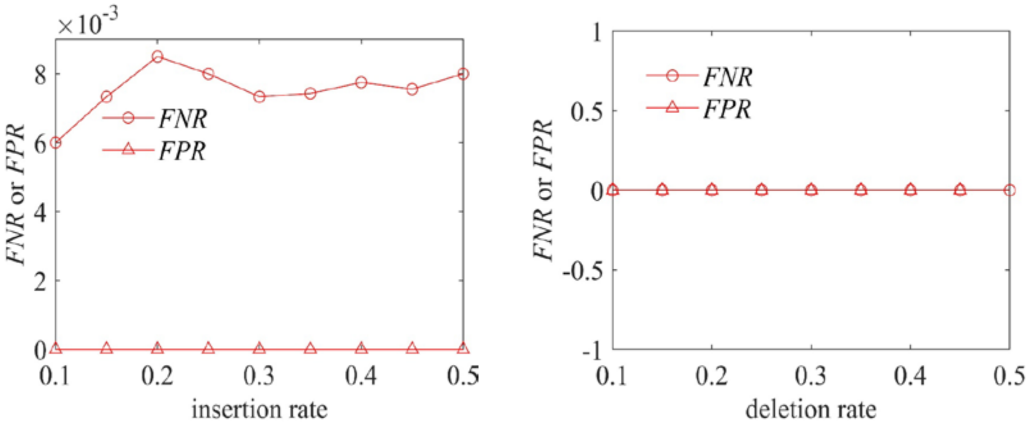


Fig. 5. *FNR* and *FPR* tests for the three cases of modification attacks. (a) Test results for the first case of modification attack, (b) Test results for the second case of modification attack, (c) Test results for the third case of modification attack.

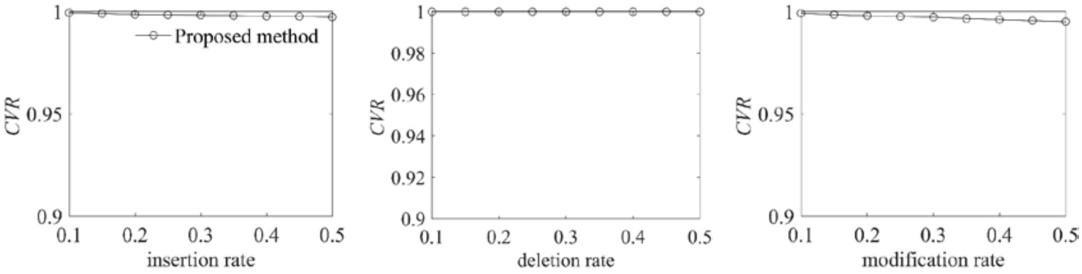
Modifications to different parts of the data are divided into three cases to illustrate. In the first case, the data used to generate watermark is only modified. As seen in **Fig. 5** (a), *FPR* remains at 0 and *FNR* approaches 0.02 when the modification rate increases. In the second case, the watermark part of the data is only modified. **Fig. 5** (b) shows that the *FNR* and *FPR* always stay as 0. In the third case, both data used to generate watermark and watermark part are modified. In this case, the generation watermark and the extraction watermark for part of the data is consistent. It can be seen from **Fig. 5** (c) that the value of *FNR* is between the first and second cases.

Fig. 6 (a-b) shows the experimental results of data insertion and data deletion. At different insertion rates, *FNR*s of the scheme were greater than 0 and less than 0.015. For all different insertion rates, the *FNR*s of the proposed scheme are larger than 0 and lower than 0.015. The *FNRs* values are 0 for all different deletion rates. Results explained in **Fig. 6** indicate a good performance for the proposed scheme in verifying data integrity with different tampering attacks.



(a) (b)

Fig. 6. *FNR* and *FPR* tests for insertion and deletion attack. (a) Test results for the insertion attack, (b) Test results for the deletion attack.



(a) (b) (c)

Fig. 7. *CVR* tests for tampering attacks. (a) Insertion attack test, (b) Deletion attack test, (c) Modification attack test.

It can be seen from **Fig. 7** (a, c) that for all the different insertion rates and modification rates, *CVR* gradually decreases as the modification rate and insertion rate increase. At the same time, the accuracy of verifying the integrity of the tampered data exceeds 0.99. It can be observed from **Fig. 7** (b) that the results of the *CVR* is always equal to 1. The reason is that if a data item is deleted during transmission, the receiving end cannot verify the integrity of the deleted data and can only verify the received data items. When data is subject to tampering attacks during transmission, the proposed scheme will detect each data item. *CVR* means verifying the integrity of all received data. In the verification process, because there is no data being discarded, this scheme has a good advantage in verifying individual data. In addition, it can be seen that the proposed scheme has good performance in verifying the integrity of all test data.

4.2 Performance evaluation comparison among individual schemes

For experimental comparison, the data stream is modified with an attack rate *r*. Next, about 10000*r* data are tampered in each experimental data stream. The corresponding *FNR* and *FPR* values highlight the ability of various schemes for identifying 10000*r* tampered data. For a fair comparison, the method in [27] which the LSB ratio *p*=0.3, the threshold *T*=4, and queue length *N*=60. In [28], the threshold *M*=4, group length *N*=70, and LSB ratio *pp*=0.2.

4.2.1 Comparison of tamper detection

Fig. 8 shows that the *FPRs* for schemes reported in [27] and [28] are more than 0.8, and approach 1 as the modification rate increases. The *FNRs* for scheme in [27] are around 0, while for scheme in [28] are between 0-0.2. However, *FNR* of the proposed scheme is smaller and more stable when compared to other two schemes. Furthermore, in the schemes proposed in [27] and [28], for related data items, the watermark generated by them is consistent. Hence, when one of the data items is tampered with, authentication of the other related non-tampered data items fails. Consequently, data-independent authentication has a higher tamper detection rate than data-associated authentication.

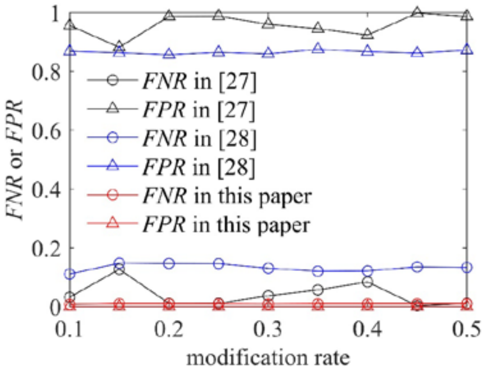


Fig. 8. Comparison among individual schemes.

In addition, queue caching data consumes more storage room in the scheme proposed in [27]. By contrast, the proposed scheme only caches one data item, hence having lower resource consumption. To sum up, it can be clearly observed from **Fig. 8** that the scheme proposed in this work outperforms the compared schemes from literature, in terms of performance of data integrity verification.

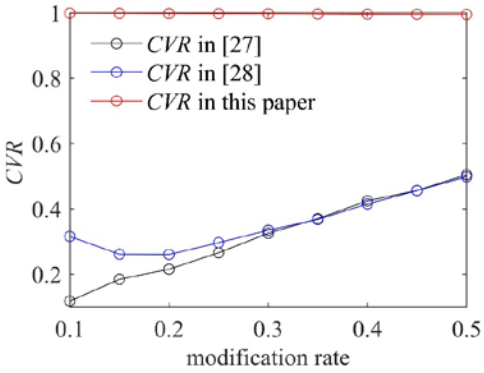


Fig. 9. *CVR* comparison among individual schemes.

It can be seen from **Fig. 9** that the *CVR* of the scheme [27] and scheme [28] increases with the increase of the modification rate. As the modified data increases, the total number of correct data will decrease. The changing trend of the false positive rate is relatively stable, so the number of misjudged data will be significantly reduced. It can be seen that the *CVR* will increase as the modification rate increases. The *CVR* of the proposed scheme is stable and close to 1 under the change of the modification rate. Because the proposed scheme has no false positive, the trend of *CVR* changes is only related to false negative.

From **Fig. 10**, it can be seen that, under the same conditions, encoding computation time of the proposed scheme is generally equal to that of scheme proposed in [28] and much lower than reported in scheme [27]. Since the data used to generate watermark in the scheme proposed in [27] repeatedly moves in and out of the queue, it consumes large time on computation. Although the computation time of the scheme [28] is lower than that of the proposed scheme when the data stream is 8000, the computation time of the proposed scheme is even shorter when the data stream reaches 10000. Overall, the computation time of the proposed scheme and scheme [28] is similar. Furthermore, the scheme proposed in [28] groups the data, which increases the overhead and has higher *FNR* and *FPR*. Conclusively, the proposed scheme improves the performance of data integrity verification without increasing the computational costs.

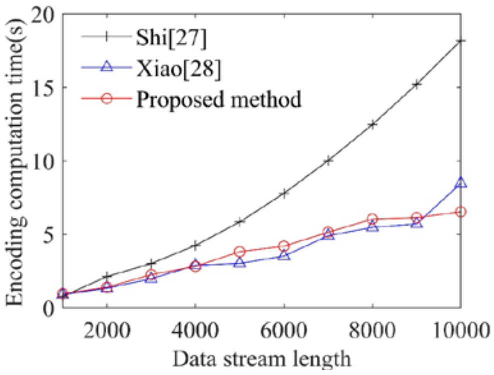


Fig. 10. Comparison of encoding computation time.

4.2.2 Computation and communication overhead

In the proposed scheme, the security of data is considered at the network layer, and the data collected from sensor nodes are sent to the sink node for verification. Because the computing resources of the base station are sufficient, the resource consumption of the sensing node is mainly compared. This subsection mainly analyzes the computational overhead, storage overhead and transmission energy consumption of the proposed scheme and the other two individual schemes at the sensor nodes. For the communication overhead, in the research results of the literature [31], the node consumes 0.6 nJ of energy per sending 1 bit of data, and consumes 0.67 nJ of energy per 1 bit of data received. **Table 3** presents the simulated parameters and settings to evaluate the performance of different schemes.

Table 3. Simulation parameter settings

Simulation parameter options	Parameter setting values
Target zone	100×100
Number of Nodes	21
Data-collection interval	0.001s
Computational energy consumption instruction	0.6nJ

The proposed scheme and Shi [27] and Xiao [28] mainly use the hash function to generate watermark. Although the computational cost of the hash function is relatively large compared to other operations, the hash function is a relatively low-cost function in the encryption method. Meanwhile, the literature [32] proved that hash function can be applied to WSN. In addition, the proposed scheme is simple arithmetic operation, logical operation and bit operation except hash operation. The overhead of generating a watermark includes multiplying, adding, and XOR operation. The process of embedding watermark is mainly two operations of converting three bits into decimals, including multiplication and addition. The proposed scheme does not need to cache data items in the sensor node, as long as the sensing data of the current data item is embedded with watermark and the data is sent out immediately. Therefore, there is no memory overhead in this scheme.

For the scheme proposed by Shi [27], firstly, the perceptual data items are copied and stored in a cache queue of length n, and then the matrix B_{nm} generated by probability P is used to randomly select m data for calculating watermark, which includes dot product operation. The watermark is generated by hash and XOR operation. Embedding the watermark using the LSB method only requires multiplication and addition operations. In the scheme Xiao [28], similar to Shi [27], a dot product operation is performed on the group of length n to select data items. The data is then converted into a character array and the bits are embedded in the grouped data items, respectively. The embedding operation adopts the method of adding spaces, and the process is mainly that the assignment operation can be ignored. It is generally assumed that a data record contains temperature and time, and one data record converted to binary is 50 bits. **Table 4** lists the computational overhead and memory overhead of the proposed scheme and the compared individual schemes in the sensor according to the analysis results, where M represents the multiplication operation, N represents the dot multiplication operation, A represents the addition operation, H represents a hash operation.

When considering the communication energy consumption, the time stamps of the data in each scheme are consistent, regardless of the packet length, only the size of the sent message is analyzed, and the extra overhead is not compared. The sensing node sends a data record with a watermark each time. Assuming that the data record transmitted by Shi [27] is a decimal, the size of the sent message is 50 bits. Since the data records transmitted in the Xiao [28] scheme are character types, the transmitted information needs to be converted from characters to corresponding integers and then to binary, and spaces are added to some character data, so a total of 67 bits need to be transmitted. The proposed scheme is similar to Shi [27], the information transmitted by a node each time is mainly a data item with a watermark, and the watermark is embedded in the data and no additional information needs to be transmitted. **Table 5** mainly compares the energy consumed by the three schemes in transmitting information.

Table 4. Comparison of computation and communication costs of three methods in sensor nodes

Schemes	Computation cost	Extra memory overhead
Shi [27]	N+M+H+A+XOR	*n*×(50 bits)
Xiao [28]	N+H+XOR	*n*×(67 bits)
Proposed method	M+A+H+XOR	50 bits

Table 5. Energy consumption comparison

Schemes	Energy consumption
Shi [27]	30 nJ
Xiao [28]	40.2 nJ
Proposed method	30 nJ

It can be seen from **Table 5** that the proposed scheme is similar to scheme of Shi [27] in transmission energy consumption, but better than scheme of Xiao [28]. In general, the proposed scheme is better than the comparison scheme in energy consumption and computational overhead. Since there is no redundant grouping and encryption overhead, the proposed scheme is very suitable for application in wireless sensor networks.

4.3 Comparison among grouping schemes

The comparison between the grouping scheme and the proposed scheme is shown in **Table 6**, where the tampering rate is set to 20%. The table analyzes the advantages and disadvantages of grouping schemes and individual schemes through four indicators. Comparison of different watermarking methods can reflect the performance of different schemes for integrity verification.

Table 6. Comparison between other grouping schemes and proposed schemes

Method	Encryption	Detection accuracy rate	Watermark Technology	Discard
Guo [17]	No	40%	Hash	Yes
Jiang [26]	Yes	80%	Differential expansion	Yes
Shi [29]	No	98%	Differential expansion	Yes
Proposed	No	99%	Hash	No

In the grouping scheme, if one of the nodes is attacked, the packets related to the node will be discarded, which will result in packet loss and misidentification of data. Moreover, the proposed scheme can identify all data items. It can be seen from **Table 6** that the proposed scheme has a higher recognition rate of data tampering than the grouping scheme.

5. Conclusion

In this paper, both watermark generation and embedding for authentication of data are studied based on individual data items. As a result, the proposed scheme can effectively verify the integrity of the data under tampering attacks, while reducing the overall false negative rate by an average of 5% and avoiding false positives. Compared with other reported individual schemes, the proposed scheme has higher data integrity recognition ability and increases the correct verification rate by 50% on average. In addition, no flag bits and buffer areas are required for data processing, which reduces computational and communication overhead**.**

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61662039, in part by the Jiangxi Key Natural Science Foundation under Grant 20192ACBL20031, in part by the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology (NUIST) under Grant 2019r070, in part by the Graduate Scientific Research Innovation Program of Jiangsu Province under Grant no. KYCX21_1016.

References

[1] K. Jawad, K. Mansoor, A. F. Baig, A. Ghani and A. Naseem, “An Improved three-factor anonymous Authentication Protocol for WSNs based IoT System Using Symmetric cryptography,” in *Proc. of 2019 International Conference on Communication Technologies (ComTech)*, pp. 53-59, 2019. Article (CrossRef Link)

[2] F. Ishmanov, S. W. Kim, “A novel trust establishment method for wireless sensor networks,” *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 9, no. 4, pp. 1529-1547, 2015. Article (CrossRef Link)

[3] M. Boulou, T. Yélémou, D. A. Rollande and H. Tall, “DEARP: Dynamic Energy Aware Routing Protocol for Wireless Sensor Network,” in *Proc. of 2020 IEEE 2nd International Conference on Smart Cities and Communities (SCCIC)*, pp. 1-6, 2020. Article (CrossRef Link)

[4] A. M. Khedr and P. R. P V, “An Energy Efficient Data Gathering Protocol for Heterogeneous Mobile Wireless Sensor Networks,” in *Proc. of 2020 17th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 366-371, 2020. Article (CrossRef Link)

[5] J. Chen, T. Li, J. Wang and C. W. d. Silva, “WSN Sampling Optimization for Signal Reconstruction Using Spatiotemporal Autoencoder,” *IEEE Sensors Journal*, vol. 20, no. 23, pp. 14290-14301, 2020. Article (CrossRef Link)

[6] H. A. Maw, H. Xiao, B. Christianson and J. A. Malcolm, “BTG-AC: Break-the-Glass Access Control Model for Medical Data in Wireless Sensor Networks,” *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 3, pp. 763-774, 2016. Article (CrossRef Link)

[7] G. Dinc and O. K. Sahingoz, “Smart Home Security with the use of WSNs on Future Intelligent Cities,” in *Proc. of 2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)*, pp. 164-168, 2019. Article (CrossRef Link)

[8] S. Verma, P. Pillai and Y. F. Hu, “Energy-efficient privacy homomorphic encryption scheme for multi-sensor data in WSNs,” in *Proc. of 2015 7th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1-6, 2015. Article (CrossRef Link)

[9] T. D. Engouang and L. Yun, “Aggregate over multi-hop homomorphic encrypted data in wireless sensor networks,” in *Proc. of 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, pp. 248-252, 2013. Article (CrossRef Link)

[10] J. Lokesh and E. Munivel, “Design of robust and secure encryption scheme for WSN using PKI (LWT-PKI),” in *Proc. of 2009 First International Communication Systems and Networks and Workshops*, pp. 1-2, 2009. Article (CrossRef Link).

[11] M. A. Al Sibahee, S. Lu, Z. A. Hussien, M. A. Hussain, K. A. Mutlaq and Z. A. Abduljabbar, “The Best Performance Evaluation of Encryption Algorithms to Reduce Power Consumption in WSN,” in *Proc. of 2017 International Conference on Computing Intelligence and Information System (CIIS)*, pp. 308-312, 2017. Article (CrossRef Link)

[12] W. S. Aldolimi, A. A. Hnaif and M. A. Alia, “Light Fidelity to Transfer Secure Data Using Advanced Encryption Standard Algorithm,” in *Proc. of 2021 International Conference on Information Technology (ICIT)*, pp. 963-967, 2021. Article (CrossRef Link)

[13] L. Harn, C. -F. Hsu, Z. Xia and Z. He, “Lightweight Aggregated Data Encryption for Wireless Sensor Networks (WSNs),” *IEEE Sensors Letters*, vol. 5, no. 4, pp. 1-4, 2021. Article (CrossRef Link)

[14] F. H. Kumbhar, N. Saxena, A. Roy, “Social Reliable D2D Relay for Trustworthy Paradigm in 5G Wireless Networks,” *Peer-to-Peer Networking and Applications*, vol. 13, no. 5, pp. 1526–1538, 2020. Article (CrossRef Link)

[15] T. Wang, L. Wang and X. Wu, “CQDW: A Cyclic-Queue-based Dynamic Watermarking Mechanism for WSNs,” in *Proc. of 2019 Chinese Automation Congress (CAC)*, pp. 4052-4056, 2019. Article (CrossRef Link)

[16] D. E. Boubiche, S. Boubiche and A. Bilami, “A Cross-Layer Watermarking-Based Mechanism for Data Aggregation Integrity in Heterogeneous WSNs,” *IEEE Communications Letters*, vol. 19, no. 5, pp. 823-826, 2015. Article (CrossRef Link)

[17] H. Guo, Y. Li, and S. Jajodia, “Chaining watermarks for detecting malicious modifications to streaming data,” *Information Sciences*, vol. 177, no. 1, pp. 281-298, 2007. Article (CrossRef Link)

[18] K. Hameed, I. Ahmed, Z. U. Ahmad, et al., “A Zero Watermarking Scheme for Data Integrity in Wireless Sensor Networks,” in *Proc. of 2016 19th International Conference on Network-Based Information Systems (NBiS)*, pp. 119-126, 2016. Article (CrossRef Link)

[19] G. Zhang, L. Kou, L. Zhang, et al., “A new digital watermarking method for data integrity protection in the perception layer of IoT,” *Security and Communication Networks*, vol. 2017, 2017, Article ID 3126010. Article (CrossRef Link).

[20] T. -M. Hoang, V. -H. Bui, N. -L. Vu and D. -H. Hoang, "A Lightweight Mixed Secure Scheme based on the Watermarking Technique for Hierarchy Wireless Sensor Networks," in *Proc. of 2020 International Conference on Information Networking (ICOIN)*, pp. 649-653, 2020. Article (CrossRef Link)

[21] G. Padmavathi, D. Shanmugapriya and M. Kalaivani, "Digital watermarking technique in vehicle identification using wireless sensor Networks," in *Proc. of 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, pp. V2-6-V2-10, 2010. Article (CrossRef Link)

[22] T. -S. Chen, K. -N. Hou, W. -K. Beh and A. -Y. Wu, "Low-Complexity Compressed-Sensing-Based Watermark Cryptosystem and Circuits Implementation for Wireless Sensor Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 11, pp. 2485-2497, Nov. 2019. Article (CrossRef Link).=

[23] R. Bista, H. -K. Yoo and J. -W. Chang, “A New Sensitive Data Aggregation Scheme for Protecting Integrity in Wireless Sensor Networks,” in *Proc. of 2010 10th IEEE International Conference on Computer and Information Technology*, pp. 2463-2470, 2010. Article (CrossRef Link)

[24] Q. Zhou, X. Qin, G. Liu, H. Cheng and H. Zhao, “An Efficient Privacy and Integrity Preserving Data Aggregation Scheme for Multiple Applications in Wireless Sensor Networks,” in *Proc. of 2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 291-297, 2019. Article (CrossRef Link)

[25] E. Elmahdi, S. Yoo, K. Sharshembiev, Y. Kim and G. -H. Jeong, "Protecting Data Integrity for Multi-Application Environment in Wireless Sensor Networks," in *Proc. of 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 90-95, 2019. Article (CrossRef Link)

[26] W. Jiang, Z. Zhang, and J. Wu, "Reversible digital watermarking-based protocol for data integrity in wireless sensor network," *Journal on Communications*, vol. 39, no. 03, pp. 118-127, 2018. Article (CrossRef Link)

[27] X. Shi, "A Statistical Integrity Authentication Scheme without Grouping for Streaming Data," in *Proc. of 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 548-552, 2014. Article (CrossRef Link)

[28] Y. Xiao and G. Gao, "Digital Watermark-Based Independent Individual Certification Scheme in WSNs," *IEEE Access*, vol. 7, pp. 145516-145523, 2019. Article (CrossRef Link)

[29] X. Shi, D. Xiao, "A reversible watermarking authentication scheme for wireless sensor networks," *Information Sciences*, vol. 240, pp. 173-183, 2013. Article (CrossRef Link)

[30] Intel Lab Data. [Online]. Available: http://db.lcs.mit.edu/labdata/labdata.html.

[31] L. Yang, C. Ding, M. Wu. "RPIDA: recoverable privacy preserving integrity-assured data aggregation scheme for wireless sensor networks," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 12, pp. 5189-5208, 2015. Article (CrossRef Link)

[32] H. M. Al-Mashhadi, H. B. Abdul-Wahab and R. F. Hassan, "Secure and time efficient hash-based message authentication algorithm for wireless sensor networks," in *Proc. of 2014 Global Summit on Computer & Information Technology (GSCIT)*, pp. 1-7, 2014. Article (CrossRef Link)

Guangyong Gao received the Ph.D. Degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2012. He is currently a professor with the School of Computer Science, Nanjing University of Information Science and Technology. His research interests include reversible data hiding, computer networks security, multimedia information security, and digital image processing.

Min Wang received her BS degree in internet of things engineering from Wanjiang College of Anhui Normal University in 2020, China. She is currently pursuing her MS degree in software engineering at the College of Computer and Software, in Nanjing University of Information Science & Technology, China. Her research interests include digital watermark based on WSNs.