Hindawi
Security and Communication Networks
Volume 2017, Article ID 3126010, 12 pages
https://doi.org/10.1155/2017/3126010

Research Article

A New Digital Watermarking Method for Data Integrity Protection in the Perception Layer of IoT

Guoyin Zhang,[1] Liang Kou,[1] Liguo Zhang,[1] Chao Liu,[2] Qingan Da,[1] and Jianguo Sun[1]

[1] *School of Computer Science and Technology, Harbin Engineering University, Heilongjiang, Harbin, China*
[2] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*

Correspondence should be addressed to Jianguo Sun; sunjianguo@hrbeu.edu.cn

Received 19 May 2017; Revised 31 August 2017; Accepted 17 September 2017; Published 30 October 2017

Academic Editor: Rongxing Lu

Since its introduction, IoT (Internet of Things) has enjoyed vigorous support from governments and research institutions around the world, and remarkable achievements have been obtained. The perception layer of IoT plays an important role as a link between the IoT and the real world; the security has become a bottleneck restricting the further development of IoT. The perception layer is a self-organizing network system consisting of various resource-constrained sensor nodes through wireless communication. Accordingly, the costly encryption mechanism cannot be applied to the perception layer. In this paper, a novel lightweight data integrity protection scheme based on fragile watermark is proposed to solve the contradiction between the security and restricted resource of perception layer. To improve the security, we design a position random watermark (PRW) strategy to calculate the embedding position by temporal dynamics of sensing data. The digital watermark is generated by one-way hash function SHA-1 before embedding to the dynamic computed position. In this way, the security vulnerabilities introduced by fixed embedding position can not only be solved effectively, but also achieve zero disturbance to the data. The security analysis and simulation results show that the proposed scheme can effectively ensure the integrity of the data at low cost.

1. Introduction

With the rapid development of computer technology, embedded technology, Internet, and mobile communication network, IoT emerges at a historic moment. The basic characteristic of IoT is the comprehensive perception, reliable transmission, and intelligent processing of information, and the key is to realize the information interaction between human and things or things and things [1]. Since its introduction, IoT has caused great repercussions all over the world that a lot of manpower and material resources have been invested to support the research, and remarkable achievements have been obtained. The rapid growth of IoT has caused great changes in the industry, which is considered as the third wave of the world information industry following the computer and Internet [2]. China Communications Standards Association (CCSA) defines the architecture of IoT as perception layer, network layer, and application layer [3], as shown in Figure 1. The perception layer consists of sensors, RFID reader, WebCam, and smart phone which is used to perceive and collect the information of objects and the environment. The network layer consists of Internet and wireless network such as 2G, 3G, 4G, and satellite network, which is responsible for transferring the sensed data to the application layer. The application layer consists of application platform and supporting platform such as distributed parallel computing, data mining, and cloud computing. The supporting platform provides the functions for the specific application, such as data processing, data storage, and security management. The IoT application has been widely used in smart city, telemedicine, smart home, and other fields.

The network layer and application layer both apply the mature technology which will ensure the security of the sensed data. However, the perception layer consisting of simple node will face the serious security problems which attracts the attention of the majority of scholars. The main function of the perception layer is information perception. The information perception is the basis of IoT applications that provides the information from the physical world, so

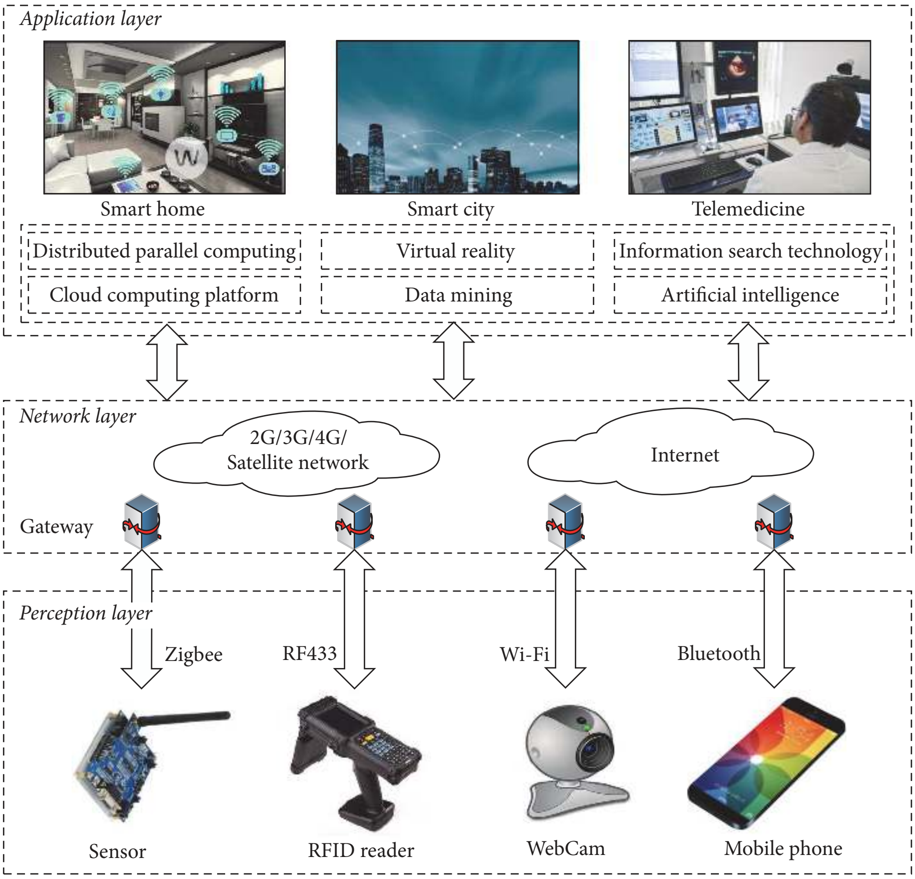


Figure 1: The architecture of IoT.

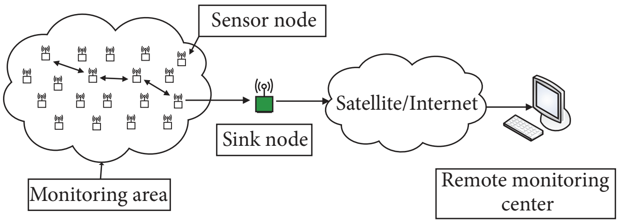


Figure 2: The communication mechanism of IoT perception layer.

the perception layer has become the main concern of the IoT related research. The communication mechanism of IoT perception layer is shown in Figure 2. The perception information plays a key role as a link between the IoT and the real world. It is composed of perception data and location data. The disclosure of perception data can lead to the disclosure of critical information across the whole network, resulting in immeasurable consequences. In this paper, we mainly consider the protection of perception data and regard the wireless sensor networks (WSNs) as the perception layer of the IoT. WSNs are self-organized by a large number of microsensors with limited computing power and small battery power to form a self-organizing network in wireless communication. The sensor data is collected from the sensor nodes and sent to the sink node by multihop relay. The sink node processes the perceptual data and sends it to the application layer through the network layer. Compared with other sensor nodes, the sink node can be regarded as a powerful computer connected to the power supply, which has fast speed processor, huge storage capacity, high network bandwidth, and security assurance [4].

Compared with traditional networks, the WSNs have some special features: (1) WSNs are data-centric networks, which focus purely on detecting and collecting the perceptual data of the sensing area, and do not care about the origin of the data. (2) The urgent problem that WSNs need to solve is to reduce the energy consumption and extend the working hours of the sensor nodes. (3) The sensor nodes are required to dynamically adapt to the changes of the network since most of them are working in harsh environment and failures can occur at any time. (4) The sensor nodes cannot perform complex calculations due to the utterly limited processor and storage capacity. (5) Security considerations should be comprehensive before focusing on the design of the WSNs, because the inherent vulnerabilities enable them easily to be attacked by various attacks, such as packet tampering attack, packet forgery attack, selective forwarding attack, packet replay attack, and transfer delay attack [5]. The data security of WSNs becomes a bottleneck that affects the further development of WSNs.

Protecting the integrity of data in WSNs is the key issue of WSNs security. Malicious modification of data can cause serious consequences. The data integrity authentication of traditional network mainly makes use of cryptography and Message Authentication Code (MAC). Although the encrypted data is safe, the data can only be used after decryption, which brings an opportunity to attackers. In addition, in order to ensure security, the encryption algorithm takes advantage of complex computational instructions that require additional space to store the keys, which undoubtedly increases a significant challenge to computational load, energy consumption, and storage space of sensor nodes [6]. The literature [7] first implements the link layer security protocol TinySec, which generates a 4-bit MAC to prevent data forgery and data tampering. Although TinySec is optimized based on the highly constrained resource of WSNs, there is still an additional payload that cannot be ignored for the sensor nodes. Considering the computing power, storage space and energy supply of sensor nodes are limited; the proposed strategies based on MAC and encryption are not applicable for the WSNs [8].

In order to solve the deficiencies of the traditional data integrity authentication method, researchers have introduced digital watermarking technology into WSNs to protect the data integrity [9]. Digital watermarking technology is widely used to protect copyright information and content integrity of multimedia digital works (images, audio and video, etc.) [10]. Compared with the traditional encryption technology, digital watermarking technology has the following four advantages: (1) the operation of watermark generation, watermark embedding, and watermark extraction uses lightweight calculations leading to low energy consumption; (2) the watermarks information is directly integrated into the carrier data without additional overhead for network communication and storage capacity of nodes in WSNs; (3) once the encrypted data is decrypted, the protection of the encryption technique loses its effect, but as the inseparable part of the host carrier, the watermarks can always guarantee the data security [11]; (4) the digital watermarking technology can significantly reduce the end-to-end delay caused by encryption technology. According to the antiattack characteristics, digital watermarking technology can be divided into fragile watermarking and robust watermarking [12]. Robust watermarking is not sensitive to modifications and can be used for copyright protection. Fragile watermarking is extremely sensitive to tampering, and any modifications to the carrier can lead to the failed extraction of watermark, which can be used to verify the integrity of data [13].

2. Related Work

This paper first introduces some of the protection mechanisms for the security of the Internet of things. Then we mainly focus on the data integrity protection of perception layer (WSNs) of the IoT. The data integrity means that the data received by the recipient is consistent with the data sent by the sender in the transmission process. Various data integrity protection strategies in WSNs have been proposed, summarized as follows.

Li et al. [14] propose the RealAlert that is a security sensing strategy based on policy for IoT. The strategy applies the reporting history and the policy rules for data collection to ensure the trustworthiness of the data and the IoT devices. Li and Song [15] propose a trust scheme for the vehicular ad hoc networks (VANETs) to protect collected data and the node in the VANETs. The proposed model can appraise the trustworthiness of the data and the nodes, respectively. What is more, it can also locate the malicious node in the VANETs and own resistance to a variety of attacks. Anbuchelian et al. [16] apply the trust mechanism for the WSNs cluster head selection. The trust mechanism named Firefly based metaheuristic will improve the security while extending the life cycle of WSNs.

Feng and Potkonjak [17] implemented the first real-time digital watermarking system in WSNs to validate the integrity of sensed data. The proposed algorithm embedded the encryption code of digital signature into the sensed data collected by WSNs. Taking the advantage of the property of sensor nodes, which allowed the existence of certain error of various practical parameters, the watermark information was embedded by modifying the value of actual parameters within allowable range. Taking the positioning process of atomic triangulation as an example, it transformed the localization problem into the optimal solution of the nonlinear equation and embeds the encrypted signature of author into the coefficients of equation. However, the limitation of the optimal solution of nonlinear equations led to the fact that this method cannot be widely used.

Guo et al. [18] proposed a new fragile digital watermarking algorithm SGW, which could verify the integrity of the data stream from the application layer. The literature grouped the data according to the key and calculated the hash value of data from each group as the watermark. Then the watermark was directly embedded into the Least Significant Bit (LSB) of the data from each group to save bandwidth. The proposed method used watermark to link all groups, to detect deletion of the data and even the whole group. However, the design of the strategy did not take limited energy supply and computing power of WSNs node into account, so it could not be applied to WSNs directly. Kamel and Juma [19] proposed lightweight chained watermarking (LWC) to optimize the SGW and achieved high performance. It also applied dynamic group size and used the hash value of two consecutive groups of data as the watermark to save computational overhead significantly instead of calculating the hash value of each data element in the group.

Kamel and Juma [20] proposed FWC-D algorithm to address the inherent security vulnerabilities of the above two methods. The algorithm first divided the sensed data into groups of the same size according to the delimiter (the value that the sensed data could not reach). The algorithm generated a sequence number SN for each group and embedded it into the group to achieve the purpose of detecting the operation of deleting or adding group. Digital watermark was obtained by the hash function through the key K, group data, and group serial number. In order to avoid replay attacks, the algorithm embedded the watermark of current group into the previous group and then chained all groups with digital watermark.

Shi and Xiao [21] proposed a new data integrity authentication algorithm for WSNs based on reversible digital watermarking. The proposed algorithm effectively applied prediction-error expansion to avoid the loss of the sensed data due to embedding watermark. However, this algorithm required not only to calculate the size of the group according to the prediction function but also to calculate the hash value of each data item in the group, which greatly increased the computational complexity, so it is not suitable for the highly resource-constrained WSNs. In addition, the watermark embedding process that used the predictions of spread-error expansion might cause data underflow and overflow.

Wang et al. [22] proposed a multimarked fragile digital watermarking algorithm based on integrity of character data to detect the malicious tampering by an attacker. The algorithm used chain watermark of dynamic group size. It firstly transferred sensed data from numerical type into character type and then used the watermark embedding strategy based on blank characters. In this algorithm, the communication bandwidth and node storage capacity could be saved effectively while the data integrity was protected. However, the limited number of blank characters resulted in limited watermark capacity.

Kamel et al. [23] proposed a new lossless digital watermarking algorithm that was suitable for WSNs to verify the integrity of sensed data. Firstly, the algorithm divided the sensed data into fixed-size group and then used the variable-base factorial number system to rearrange the location of the sensed data. Secondly, it embedded the watermark information through the new order of sensed data in the group; finally, the sender and the receiver added a mapping string for each group to provide the basis for reconstructing the original data element and then extracted the watermark information. The algorithm did not bring any loss to the sensed data and could be applied to WSNs effectively due to its low computational complexity.

Sun et al. [24] proposed a lossless digital watermarking strategy which used redundant space of data to embed watermark information. The sensed data collected by the sensor nodes was repackaged; unlike the previous methods, the embedding of the watermark did not cause any modification to the original sensed data. However, this strategy still had a certain security vulnerability, because the initial value of reservation bit for watermark was zero, and the watermark embedding position was relatively fixed which could be used by the attacker to obtain the sensed data.

In conclusion, the existing WSNs data integrity verification algorithms are based on digital watermarking mostly by replacing the LSB of the sensed data with watermark data to achieve the goal of embedding watermark information. This kind of algorithm is easy to implement and has a low time complexity, which satisfies the requirements of highly resource-constrained WSNs to a certain extent. Besides, a large number of scholars put forward many improved LSB algorithms: the algorithm based on the LSB group, the algorithm based on distributed LSB, and the algorithm based on multiflag LSB. To a certain extent, although the improved algorithm enhances the security, the watermark embedding position can become a serious security vulnerability, which is prone to malicious use by the attacker. Besides, LSB will cause data damage to some extent, which is unacceptable in sensitive areas such as medical and military. What is more, in the existing algorithms, the data integrity can only be verified after the arrival of entire group of data, which leads to greater delay. In order to solve these problems, this paper presents a lightweight watermarking technique called position random watermark (PRW). The proposed algorithm calculates the embedding position of watermark dynamically by using the data collecting time from sensor nodes, which not only improves the security, but also saves energy and realizes real-time data authentication.

3. Attack Models and Proposed Scheme

3.1. The Attack Models. Compared to wired networks, the WSNs that deploy in the extreme environment face more threats, and what is more, the public communication protocol adopted by WSNs exacerbates the risk of physical tampering. The sensed node owns limited computational capabilities and energy resources which increase the difficulty of designing security protocols. We summarize the main attack models into five categories:

(a) Packet tampering: a malicious node added to WSNs tampers with the value of the packets and forwards the tampered packets which can lead to extremely serious consequences in some special cases.

(b) Packet forgery: a malicious node added to WSNs keeps sending the fake packets to other nodes, greatly increasing network traffic and resulting in wasting energy of the whole WSNs.

(c) Selective forwarding: a malicious node added to WSNs deletes partial packets and forwards some packets to destination selectively. The data loss may cause the bad situation that the sink node fails to make the correct response.

(d) Packet replay: a malicious node added to WSNs forwards the packets that have been forwarded, once more or repeatedly to other nodes which will cause the traffic congestion and energy waste.

(e) Transfer delay: a malicious node added to WSNs forwards the packets later than the predetermined time which will lead to the fact that the sink node drops the packets due to the timestamp.

3.2. The Proposed Scheme. This paper proposes a new WSNs data integrity protection strategy based on fragile digital

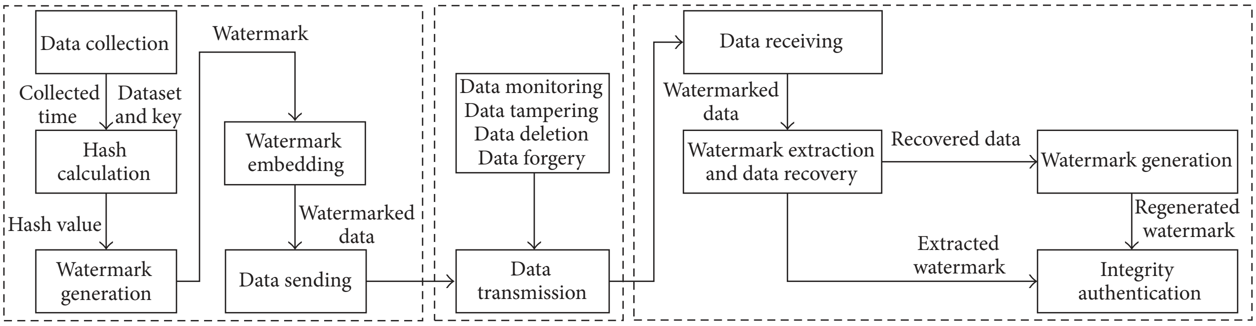


Figure 3: The implementation model of the proposed fragile watermark algorithm.

watermarking to protect the sensed data from the above four categories of attack models. The proposed algorithm makes use of the characteristic that the fragile watermarking is sensitive to modification. Once the host data is modified, the watermark is destroyed. The malicious node without the prior knowledge of watermarking algorithms cannot effectively restore real data. Data tampering and data forgery are similar, which can be seen as malicious data generated by malicious nodes. The malicious sensed data generated by the malicious node cannot be verified by watermarking algorithm after reaching the sink node. The proposed algorithm introduces the packet sequence number SN which is used for positioning the added packet or deleted packet.

The proposed watermarking algorithm includes three processes, namely, digital watermark generation, digital watermark embedding, and digital watermark extraction as shown in Figure 3: firstly, each sensing node collects the sensing data and generates the digital watermark according to the fragile watermarking algorithm. Secondly, the watermark is merged into the sensed data through the predefined rule to form a data packet that is transferred to the sink node through the transmission node. The packet may suffer from an unreliable transmission and face different kinds of attacks. Thirdly, the sink node receives the data and then extracts the watermark and restores the sensed data according to the predefined rule. The restored data is used to generate watermark according to the same algorithm. The data integrity is verified by comparing the regenerated watermark and the extracted watermark. If the regenerated watermark is not the same as the extracted watermark, the data is proved to be tempered during transmission. Otherwise, the data is proved safe. The digital watermark is copied with the copy of the digital media, and the process is hidden. If the predefined method is not known, the digital watermark is difficult to detect.

The WSNs are simplified in this paper, and only three types of nodes are considered:

(i) Sensing node is responsible for collecting data of monitoring area.

(ii) Transmission node is responsible for transferring the data to the sink node by multihop relay.

(iii) Sink node is responsible for receiving the sensed data sent by sensor node.

Table 1: Notations and parameters of algorithm.

Notation	Description
Hash()	The given one-way hash function SHA-1
‖	The concatenation operator
rand()	The random position function
t	The data collecting time
T	The time queue stored in the sink node
m	The length of watermark embedding bits
K	The secret key
data_j	The jth sensed data element
SN	The serial number inserted in each packet
W	Watermark to be inserted
W'	The extracted watermark
W''	The regenerated watermark

This paper makes the following assumptions: the number of sensing nodes is n; the same sensing node defined as node exists near the sink node and uses the same protocol and parameter as the node in the sensing area; each sensing node collects sensed data at the same time in one working cycle. In order to improve the security, the collected time of the sensed data is not transmitted. The node near the sink node sends the time for collected sensed data to the sink node every working cycle. The sink node sets up a queue to save the time according to the receiving order. Table 1 shows the main notations and parameters used in the proposed algorithm.

Definition 1. In WSNs, the sensed data is encapsulated as a package according to a predefined order before transmission. A series of sensed data collected at each working cycle is defined as $s_i = \{\text{data}_1, \text{data}_2, \dots, \text{data}_n\}$, data_j represents the value of one sensed data ($j = 1, 2, \dots, n$), i represents the working cycle of the sensing node ($i = 1, 2, \dots, k$), and the working cycle i is saved to the packet sequence number SN.

Definition 2. A transmitted packet is denoted as $\text{Packet} = \{\text{Head}, \text{SN}, s_i\}$. It consists of a fixed data header, packet sequence number SN, and a series of sensed data elements s_i. The data of the packet is stored in binary mode. The watermarked packet is denoted as $\text{Packet}W$. The received packet is denoted as $\text{Packet}W'$.

```
**Input**: input parameters Packet, $t, m, K$
**Output**: Watermark $W$
for $i = 1$ to $n$ do
    Data ≔ Data ‖ Packet · data$_i$;
Data ≔ Data ‖ $t$;
Data ≔ Data ‖ $K$;
$W_0$ = Hash(Data);
$W$ = MSB($W_0, m$);
return $W$;
```

Algorithm 1: Watermark generation algorithm.

3.2.1. Watermark Generation Algorithm. The watermark generation algorithm uses the SHA-1 hash function to calculate the hash value. SHA-1 hash function not only guarantees data integrity, but also has a lightweight feature that uses 65% less memory than other hash algorithms, such as the MD5 algorithm, which is more suitable for resource-constrained WSNs [25]. The secret key K [26] is the specific information that is only known to the sender and the receiver.

The watermark generation process is described as follows:

(i) Concatenate all the sensed data elements, collected time t, and secret key K to data:

$$\text{Data} = \text{data}_1 \parallel \text{data}_2 \parallel \cdots \parallel \text{data}_n. \tag{1}$$

(ii) Calculate the hash value of the variable data denoted as W_0, based on SHA-1 hash function:

$$W_0 = \text{Hash}\left(\text{Data} \parallel t \parallel K\right). \tag{2}$$

(iii) Select m bits from most significant bits of W_0 as the watermark W, according to the actual needs.

$$W = \text{MSB}\left(W_0, m\right). \tag{3}$$

The detailed operation steps of watermark generation algorithm are shown in Algorithm 1.

3.2.2. Watermark Embedding Algorithm. The proposed watermark embedding algorithm improves from the following two aspects:

(i) The packet is redesigned and added m bits for watermark to ensure that the watermark is transparently embedded in the packet. It does not cause any interference to the data and meets the high-precision requirements.

(ii) In order to solve the vulnerabilities brought by fixed embedding location, we introduce a new position random function to dynamically calculate the watermark embedding position which effectively solves potential vulnerabilities and greatly improves the security of the algorithm.

Figure 4 illustrates the generation and embedding mechanism. The watermark embedding process is described as follows:

```
**Input**: input parameters Packet, $t, m, K$
**Output**: New packet Packet$W$
Get watermark $W$ according to Algorithm 1;
$P, Q$ = rand($t, K, m$);
for $i = 1$ to $m$ do
    index ≔ $P_i$;
    Packet$W_{\text{index}} = W_i$;
for $i = 1$ to $l$ do
    index ≔ $Q_i$;
    Packet$W_{\text{index}}$ = Packet$_i$;
return Packet$W$;
```

Algorithm 2: Watermark embedding algorithm.

(i) Calculate the watermark information W according to Algorithm 1 and update the payload of the packet from l to L.

(ii) Acquire position arrays P and Q through function rand() with collected time t, secret key K, and num of watermark bits m. P and Q represent the watermark embedding position and the sensed data embedding position, respectively. P and Q need to satisfy the formula

$$\begin{gathered} 1 \leq P_i \leq L, \quad 1 \leq i \leq m; \\ P_i \neq P_j, \quad i \neq j; \\ P_i < P_j, \quad i < j \\ 1 \leq Q_i \leq L, \quad 1 \leq i \leq l; \\ Q_i \neq Q_j, \quad i \neq j; \\ Q_i < Q_j, \quad i < j \\ P \cap Q = \emptyset, \quad P \cup Q = \{1, 2, \ldots, L\} \end{gathered} \tag{4}$$

$\text{Length}\,(P) + \text{Length}\,(Q) = L.$

(iii) Embed the watermark W according to the position array P and embed the sensed data according to the position array Q to form a new packet Packet$W$.

The detailed operation steps of watermark embedding algorithm are shown in Algorithm 2:

3.2.3. Watermark Extraction Algorithm. When the packet is transmitted to the sink node, the sink node extracts the digital watermark information and restore the sensed data. The received packet is denoted as PacketW'. The sink node and the sensing node share the secret key K. The restored packet is represented as Packet$'$.

The extraction and verification process are described as follows:

(i) Extract the serial number SN from received packet PacketW'.

(ii) Acquire the collected time t from the time queue stored in the sink node based on the SN.

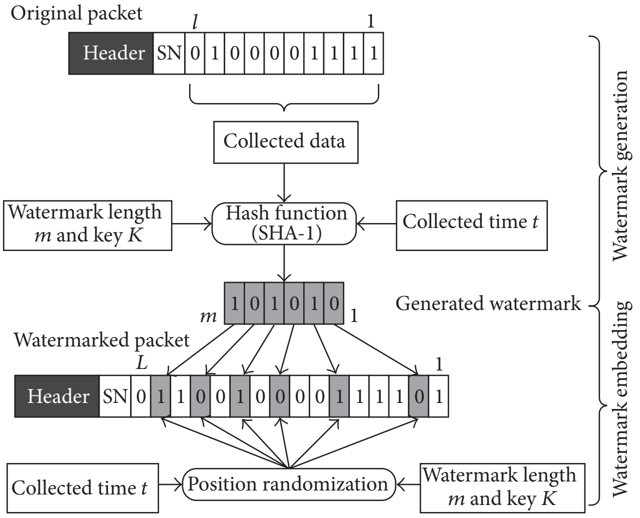


Figure 4: The processes of generation and embedding.

```
Input: input parameters Packet$W'$, $m$, $K$
Output: Integrity verification result
index ≔ Packet$W' \cdot$ SN;
$t$ ≔ time$_{\text{index}}$;
$P, Q$ = rand($t, K, m$);
for $i$ = 1 to $m$ do
    index ≔ $P_i$;
    $W'_i$ ≔ Packet$W'_{\text{index}}$;
for $i$ = 1 to $l$ do
    index ≔ $Q_i$;
    Packet$'_i$ = Packet$W'_{\text{index}}$;
for $i$ = 1 to $n$ do
    Data ≔ Data ‖ Packet$' \cdot$ data$_i$;
Data ≔ Data ‖ $t$;
Data ≔ Data ‖ $K$;
$W_0$ ≔ Hash(Data);
$W''$ ≔ MSB($W_0, m$);
if $W'' == W'$ then
    Return (Integrity intact);
else
    Return (Integrity tampered);
```

Algorithm 3: Watermark extraction algorithm.

(iii) Calculate the arrays P and Q according to the function rand().

(iv) Obtain the watermark denoted as W' and restore the packet represented as Packet$'$.

(v) Recalculate the watermark W'' according to Algorithm 1 with Packet$'$, t, m, and K.

(vi) Compare W' with W''; if W' is equal to W'', the data integrity is verified; otherwise the data is tampered.

Figure 5 illustrates the extraction and verification mechanism. The detailed operation steps of watermark extraction algorithm are shown in Algorithm 3:

4. Security Analysis

The malicious node added to the WSNs can launch various attacks based on the attack models mentioned before. Attacks usually happen during the transmission. One attack is supposed to be successful if the sink node cannot detect the modification of the sensed data. In this section, we discuss how the proposed watermarking algorithm resists various attacks.

4.1. Modification

4.1.1. Modification of One Data Element. If the attacker just modifies one data element and the embedded watermark remains unchanged. The sink node can extract the right watermark information and restore the wrong data element according to the extraction algorithm. The modification of one data element can lead to the wrong hash value and the wrong recalculated watermark that does not match the extracted watermark. The sink node will reject the modified packet. If an attack modifies multiple data elements, the result is similar.

4.1.2. Modification of Embedded Watermark. If the attacker just modifies the embedded watermark and the data elements remain unchanged, this may result in the wrong extracted watermark and the right recalculated watermark that lead to the failed authentication.

4.1.3. Modification of SN. We assume the attacker modifies the serial number SN. The changed SN leads to the wrong collected time in the sink node. It affects both extraction watermark and restored sensed data because the wrong collected time can result in the wrong embedding position. The recalculated watermark and the extracted watermark are inconsistent. The sink node will reject the modified packet.

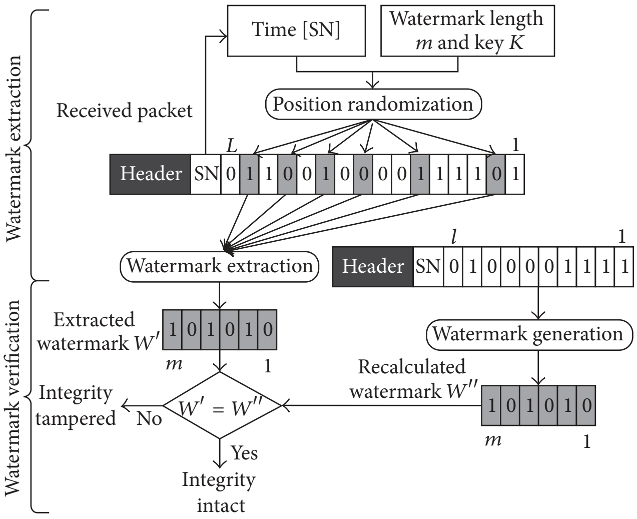


Figure 5: The processes of extraction and verification.

4.2. Insertion/Deletion

4.2.1. Insertion/Deletion of Element of Packet. When the packet arrives, the sink node checks its length. No matter one element or multiple elements are inserted into the packet, the length of packet does not meet the requirement, which result in the rejection of this packet. The deletion of element is similar to the insertion, so it is not discussed here in detail.

4.2.2. Insertion/Deletion of Whole Packet. The collected time and the secret key are known to the sender and receivers only but are not known to an attacker. If the attacker deletes one or more packets, the sink node may locate these packets according to the SN. Without the knowledge of collected time and secret key, the attacker can hardly get the correct embedded position and watermark, so the attack cannot generate the packets that meet the requirements. When the inserted packets arrive at the sink node, they cannot be authenticated successfully and are rejected.

The above analyses are sufficient to prove that the proposed algorithm can resist various attacks of WSNs, such as modification, insertion, and deletion. It also applies to a combination of scenes. It is proved that the algorithm can guarantee the integrity of data of the WSNs.

5. Experimental Results

The experimental data used in this paper is the real sensed data collected by the Intel Berkeley Research Laboratory, which contains humidity data, temperature data, light intensity data and voltage data, and the time to obtain these data. However, the collected time of different types of sensed data is not the same. In order to save the amount of computation, it is assumed that all types of data in the one working cycle use the same collected time.

We use the Network Simulator Ns-2 to implement simulation experiment that evaluates the performance of the proposed PRW algorithm. Ns-2 is the open source simulation platform that simulates discrete events. It is widely used in academia because of its scalable features. Table 2 shows the significant notations and parameters of the simulation.

Table 2: Notations and parameters of simulation.

Parameter	Value
Surface of the network	100 m ∗ 100 m
Num of sink node	1
Num of sensing node	8
Num of attacker node	10
Num of transmission node	81
Sink node location	(30, 60)
Routing protocol	LEACH
MAC protocol	S-MAC
Initial energy	2 J

In order to better conserve energy, the designed experiment uses the S-MAC protocol. The S-MAC protocol is designed for the resource-constrained WSNs. It has energy-efficient features because it implements the low-duty-cycle operations and periodically listening and sleeping.

In the simulation experiment, the coverage of WSNs is set to 100 m ∗ 100 m. There are 1 sink nodes, 1 sensing node near the sink node, 8 sensing nodes, 80 transmission nodes, and 10 attacker nodes randomly distributed in the sensing area, and the total number of sensing nodes is 100. In each packet, there are 1-byte packet header, 2-byte packet SN, 1-byte redundant space, and 8 bytes for data elements. The duration of each simulation is set to 200 seconds. Each simulation randomly selects 200 packets as experimental samples. The average value of the 20 simulations is used as the result data.

We conduct two sets of experiments. In the experiment selection of m, we aim to select the right parameter m to achieve the best experimental results with the minimum calculated load. In the experiment antiattack, we verify the antiattack ability of our method against five common attacks.

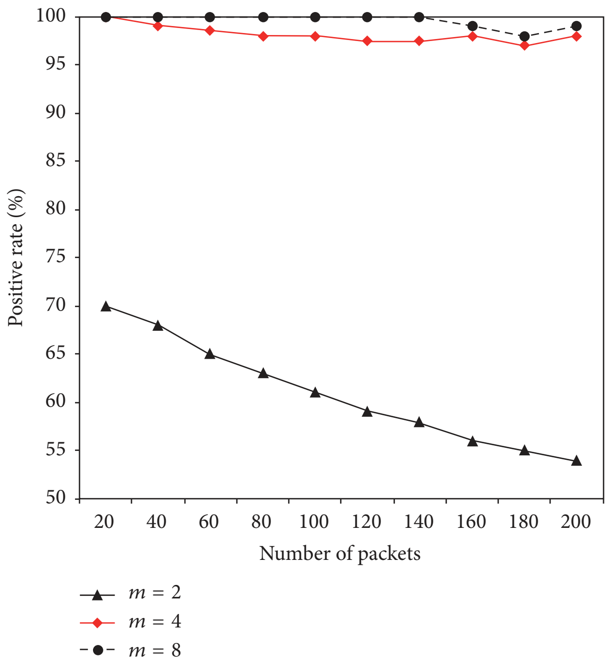


Figure 6: The detection rate under different embedding parameters *m*.

The rest of the experiments compare the embedded capacity, energy consumption, delay, and detection rate with the existing LSB methods SGW [18], LWC [19], FWC-D [20], and multimark [22], respectively, to prove that our algorithm has better performance than the baseline.

5.1. The Selection of m. The parameter m indicates the watermark bit. In order to select the appropriate parameter m that meets the safety requirements and does not lead to higher false positive rate, we select three values of the parameter m 2, 4, and 8. The experiment introduces the detection rate, which is the probability that the algorithm successfully detects the tampered packet. It is defined as

$$P_D = \frac{N_D}{N_{\text{Total}}}, \tag{5}$$

where N_D is the number of tampered packets that the extracted watermark does not match the recalculated watermark and N_{Total} is the number of all packets arriving at the sink node.

The experimental results are shown in Figure 6. It can be concluded from the experimental results that the bigger value of parameter m corresponds to higher detection rate and the algorithm owns higher security. When the parameter m falls to 2, the detection rate falls rapidly because the watermark capacity is too small to detect the tampered packet effectively. The appropriate selection of the parameter m ensures both high security and high performance of the proposed watermark algorithm.

5.2. Antiattack. The 10 attacker nodes take advantage of various attacks to verify the antiattack ability of our proposed algorithm. The five attacks, packet tampering, packet forgery, selective forwarding, packet replay, and transfer delay, are performed separately. The test of each attack is repeated 50 times, and the experimental results are showed in Table 3.

Table 3: The detection rate under various attacks.

Attack type	Experiments num	Detection rate (%)
Packet tampering	50	100
Packet forgery	50	100
Selective forwarding	50	100
Packet replay	50	100
Transfer delay	50	100

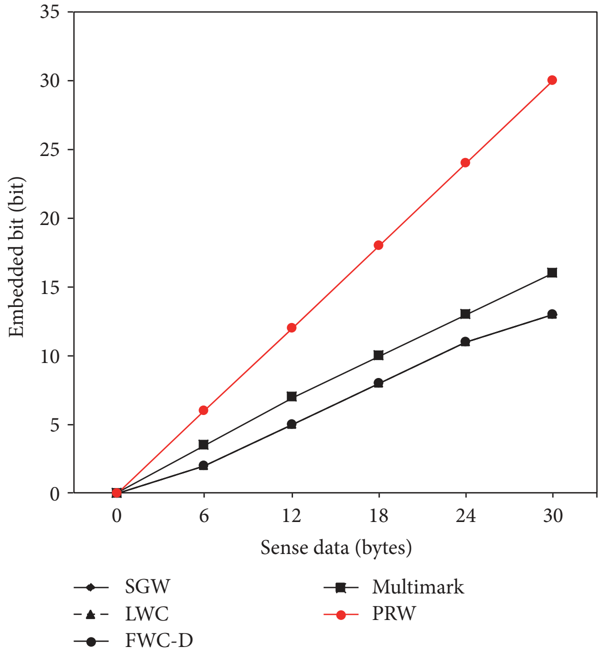


Figure 7: The watermarking embedded capacity.

According to the experimental results, our proposed algorithm effectively resists several common attacks. For the first three attacks, packet tampering, packet forgery, selective forwarding, the attack will not calculate the correct watermark information, so our watermark strategy cannot extract the correct watermark information from the packet and the data integrity certification fails. The last two attacks, packet replay and transfer delay with concealment, can evade existing programs, but our algorithm associates the serial number and collected time effectively to detect these two attacks and achieves the desired results.

5.3. Embedding Capacity. The experimental results are shown in Figure 7, whose clarity shows the superiority of the proposed algorithm in terms of embedding capacity compared to the LSB [18–20] and multimark [22]. Embedding the watermark at the lowest bit (LSB) not only limits the watermark embedding capacity, but also disrupts the integrity of the data which is fatal for the high-precision applications. Multimark [22] can guarantee data accuracy, but the number of blank characters is limited which restricts the watermark capacity.

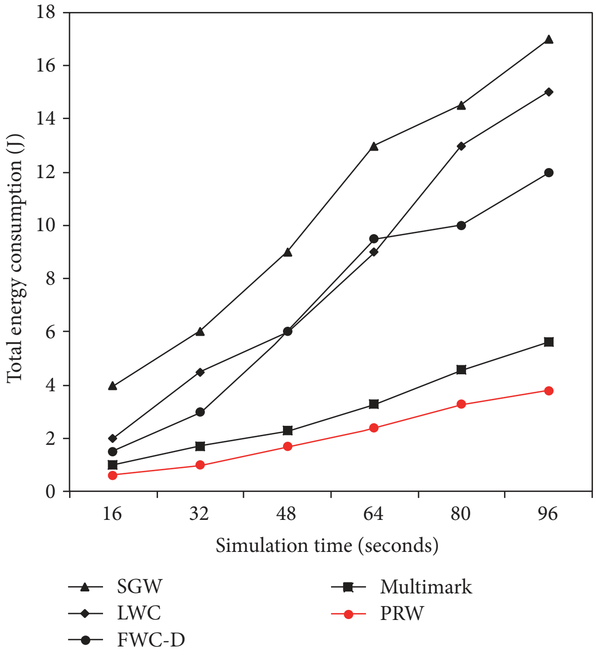


Figure 8: The energy consumption.

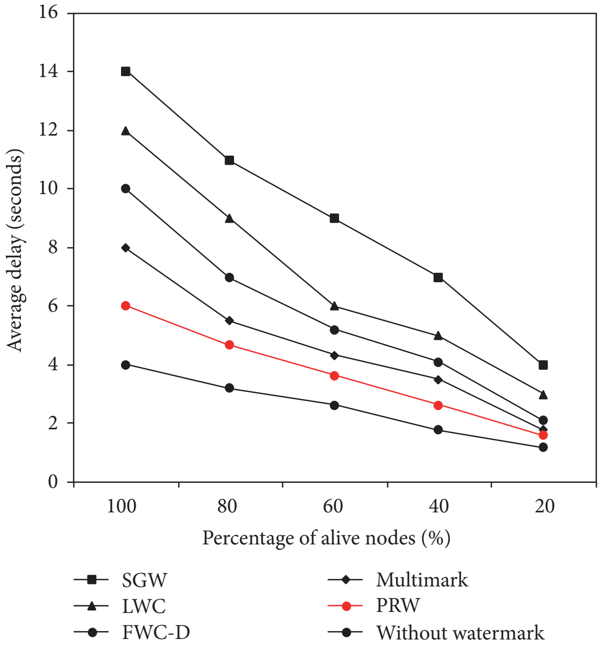


Figure 9: The average delay.

5.4. Energy Consumption. The sensed node uses the battery for energy, and the energy consumption directly determines the life cycle of the sensed node. The main energy consumption of sensed node mainly includes data collection, watermark generation and embedding, and data forwarding, where data transmission costs the highest energy.

In this paper, the bits of data packets are fixed, so the embedding of the watermark does not increase the additional storage overhead and transmission overhead. The proposed method does not cause any disturbance to the original data because the watermark is embedded into the redundant positions. The experimental results are shown in Figure 8, which clearly shows that the proposed algorithm saves more energy than the algorithms LSB [18–20] and multimark [22].

5.5. Average Delay. The delay caused by the algorithm to the WSNs mainly includes the calculation, embedding, extraction, and verification of the watermark. The proposed PRW applies the one-way hash function SHA-1 algorithm with lightweight features. Compared to other hash algorithms such as MD5, SHA-1 can calculate faster and save more storage space and energy. Therefore the watermark generation of PRW saves more time than previous watermarking schemes. Although [9] also uses SHA-1, it embeds 160 bits of data into the packet that greatly increase the transmission load. It not only consumes more energy, but also increases the transmission delay. The experimental results are shown in Figure 9, which clearly shows that the proposed algorithm reduces more delay than the algorithms proposed in LSB [18–20] and multimark [22] which will improve the response time of the network.

5.6. The Detection Rate. Due to the interference problems of wireless communication, the embedded watermark may be affected involuntarily, and the integrity authentication may fail when the packet arrived without attacks at the sink node, which is called false positive detection.

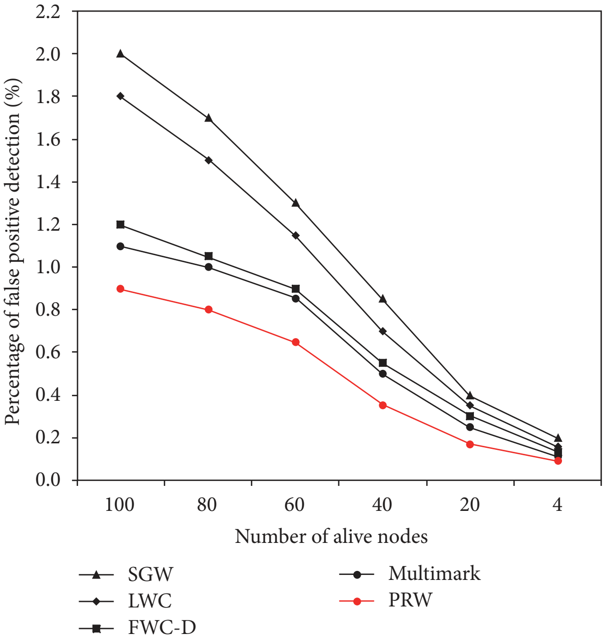


Figure 10: False positive detection.

We have conducted several simulation experiments to compare the detection rate between the PRW and previous algorithms LSB [18–20] and multimark [22]. We measure the false positive detection rate with different number of alive nodes in the network. The experimental results are shown in Figure 10, which clearly shows that the proposed algorithm decreases the false positive rate than the existing algorithms.

We can see from Figure 10 that the false positive rate is around 0.8 percent when the 80 percent sensor nodes are alive and it falls to 0.2 percent if 20 percent of nodes are alive. So we need to adjust the alert threshold based on the number of alive nodes in the WSNs dynamically.

6. Conclusion

In this paper, an advanced random digital watermarking algorithm is proposed for the data integrity of IoT perceptual layer. The proposed algorithm can effectively prevent a variety of attacks, such as packet forgery attacks, packet forwarding attacks, packet tamper attacks, packet replay attacks, and packet delay transmission attacks caused by malicious nodes. Besides, the proposed algorithm effectively solves the shortcomings of the existing technologies. It not only simplifies the computational complexity and improves the authentication efficiency and security, but also ensures the reversible extraction of watermark and the lossless restoration of data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Science Foundation of China (Grant no. 61501132, Grant no. 61370084), the China Postdoctoral Science Foundation (no. 2016M591515), the Heilongjiang Postdoctoral Sustentation Fund (no. LBH-Z14055), and Harbin Application Technology Research and Development Project (Grant no. 2016RAQXJ063, Grant no. 2016RAXXJ013).

References

[1] D. Miorandi, S. Sicari, F. de Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[2] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with China Perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349–359, 2014.

[3] J. S. Kumar and D. R. Patel, "A survey on internet of things: security and privacy issues," *International Journal of Computer Applications*, vol. 90, no. 11, pp. 20–26, 2014.

[4] A. K. Pathan, H.-W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT '06)*, vol. 2, pp. 1043–1048, Phoenix Park, Ireland, February 2006.

[5] X. Dong and X. Li, "An authentication method for self nodes based on watermarking in wireless sensor networks," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2009*, Beijing, China, September 2009.

[6] B. Przydatek, D. Song, and A. Perrig, "SIA: secure information aggregation in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 255–265, ACM, November 2003.

[7] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, November 2004.

[8] X. Ren and H. Yu, "Security mechanisms for wireless sensor networks," *Journal of Computer Science and Network Security*, vol. 6, no. 3, pp. 155-156, 2006.

[9] D. E. Boubiche, S. Boubiche, H. Toral-Cruz, A.-S. K. Pathan, A. Bilami, and S. Athmani, "SDAW: secure data aggregation watermarking-based scheme in homogeneous WSNs," *Telecommunication Systems*, vol. 62, no. 2, pp. 277–288, 2016.

[10] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proceedings of the IEEE International Conference Image Processing (ICIP '94)*, vol. 2, pp. 86–90, Austin, Tex, USA, November 1994.

[11] C. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of secure watermark-based authentication systems," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 43–55, 2006.

[12] I. Cox, M. Miller, J. Bloom, and J. Fridrich, *Digital Watermarking and Steganography*, Morgan Kaufmann, 2007.

[13] P. F. Luis, C. Pedro et al., *Watermarking Security: A Survey, Transactions on Data Hiding and Multimedia Security I*, Springer, Berlin Heidelberg, 2006.

[14] W. Li, H. Song, and F. Zeng, "Policy-based secure and trustworthy sensing for internet of things in smart cities," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1-1, 2017.

[15] W. Li and H. Song, "ART: an attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 960–969, 2016.

[16] S. Anbuchelian, S. Lokesh, and M. Baskaran, "Improving security in Wireless Sensor Network using trust and metaheuristic algorithms," in *Proceedings of the 3rd International Conference on Computer and Information Sciences, ICCOINS 2016*, pp. 233–241, August 2016.

[17] J. Feng and M. Potkonjak, "Real-time watermarking techniques for sensor networks," in *Proceedings of the Security and Watermarking of Multimedia Contents V*, pp. 391–402, January 2003.

[18] 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.

[19] I. Kamel and H. Juma, "Simplified watermarking scheme for sensor networks," *International Journal of Internet Protocol Technology*, vol. 5, no. 1-2, pp. 101–111, 2010.

[20] I. Kamel and H. Juma, "A lightweight data integrity scheme for sensor networks," *Sensors*, vol. 11, no. 4, pp. 4118–4136, 2011.

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

[22] B. Wang, X. Sun, Z. Ruan, and H. Ren, "Multi-mark: Multiple watermarking method for privacy data protection in wireless sensor networks," *Information Technology Journal*, vol. 10, no. 4, pp. 833–840, 2011.

[23] I. Kamel, O. A. Koky, and A. A. Dakkak, "Distortion-free watermarking scheme for wireless sensor networks," in *Proceedings of the International Conference on Intelligent Networking and Collaborative Systems, INCoS 2009*, pp. 135–140, November 2009.

[24] X. Sun, J. Su, B. Wang, and Q. Liu, "Digital watermarking method for data integrity protection in wireless sensor networks," *International Journal of Security and Its Applications*, vol. 7, no. 4, pp. 407–416, 2013.

[25] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, "Analyzing and modeling encryption overhead for sensor network nodes," in *Proceedings of the Second ACM International Workshop on Wireless Sensor Networks and Applications, WSNA 2003*, pp. 151–159, September 2003.

[26] M. A. Simplício Jr., P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed Wireless Sensor Networks," *Computer Networks*, vol. 54, no. 15, pp. 2591–2612, 2010.

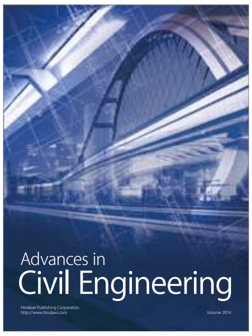
Journal of
Engineering

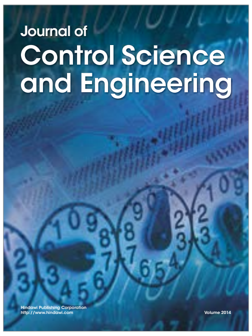
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

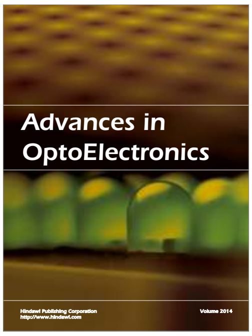
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

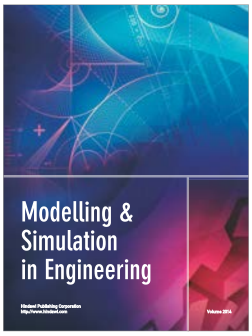
Journal of
Robotics

Journal of
Electrical and Computer Engineering


Advances in
OptoElectronics

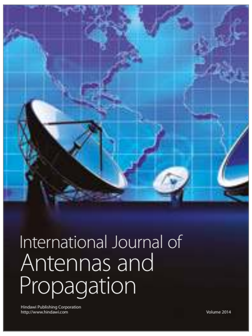
VLSI Design

International Journal of
Navigation and Observation

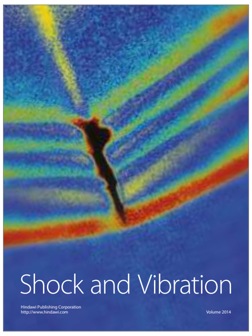

Modelling &
Simulation
in Engineering

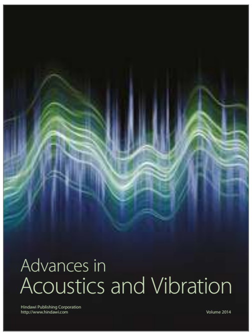
International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration