International Journal of Security and Its Applications
Vol.9, No.1 (2015), pp.125-138
http://dx.doi.org/10.14257/ijsia.2015.9.1.14

A Secure Data Transmission Scheme Based on Information Hiding in Wireless Sensor Networks

Baowei Wang[1,2],Hongwei Qian[1,2],Xingming Sun[1,2] ,Jian Shen[1,2] and Xiaoyu Xie[3]

[1]*Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, 210044, China*
[2]*School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044,China*
[3]*Wuhan Meteorological Service, Wuhan, 430000, China*
wbw.first@163.com, oneqhw@163.com, sunnudt@163.com, s_shenjian@126.com

Abstract

Wireless sensor networks hold the promise of facilitating large-scale and real-time data processing in complex environments, therefore the security is a critical issue that must be resolved. In order to prevent attacks by adversaries with fake identities, we propose a secure data transmission scheme based on information hiding technique. Firstly, one superficial change made to the beacon message of original CTP is to mark the unique identity information of a sensor node. Secondly, sensitive information is uniformly and randomly embedded into the ordinary data by using the space-efficient randomized data structure characteristic of Bloom Filter. Experimental results and performance analysis demonstrate that the proposed scheme can efficiently detect the malicious node with fake identities through the sensitive information. On the other hand, our scheme can protect the sensitive information without affecting the transmission and usability of the ordinary data.

Keywords: *Fake identities; Information hiding technique; CTP; Bloom filter*

1. Introduction

Wireless sensor networks (WSNs) typically consist of a number of wireless sensors. Data from the sensor field is collected by sensors and sent to a base station [1]. In many applications, data security is a core requirement, such as military target tracking, environmental changes monitoring and so on [2]. False or malicious data would jeopardize the entire network and finally result in incorrect decisions and potentially financial losses. Security issues in WSNs have been concerned in many literatures. Most of the prior works use traditional security solutions that are based on cryptographic algorithms [3-5]. However, it is undeniable that encrypted data might easily arouse the attention of the attacker. In particular, they usually execute thousands or even millions of multiplication instructions in order to perform operations like modular exponentiation [6, 7]. Consequently, they are too expensive and not suitable for sensors due to the limited resources of sensors.

In this case, information hiding technology is introduced to protect the data in wireless sensor networks [8-12]. Our method is hidding the encrypted sensitive information into other ordinary data by using some information hiding methods. Not only the security of the encrypted data is enhanced, but also many new benefits are brought, which are more invisible and less consumptive than cryptology.

In this paper, we propose a secure data transmission scheme to detect the malicious nodes with fake identities. First of all, we choose Collection Tree Protocol (CTP) as our data collec-

ISSN: 1738-9976 IJSIA

tion protocol in sensor networks [13]. In order to meet the requirement, we modify the structure of beacon message and routing table of CTP, which will offer condition for generating sensitive information. Secondly, by using the space efficient randomized data structure characteristic of Bloom Filter, sensitive information is randomly and uniformly embedded into the ordinary data, which will improve the anti-detection capabilities. Only the base station, who shares the pivotal parameters and the specified variation rules, can validate the complete hidden information. Any intermediate nodes of sensor network and eavesdroppers are difficult to detect the existence of hidden data. Figure 1 shows a common threat scenario in sensor networks. Node 10 is an amalicious node, it will send wrong packets or occupy the channel of other sensors. Node 5, 7, 8, 9 are within the range of its one hop communication radius. So they will get some information from the beacon message broadcasted by Node 10, which is accomplished at the route layer. Node 10 will also receive beacon messages from neighbors around, but it does not know the actual structure of the beacons. What is more, it is extremely hard to modify information at route layer from application layer in sensors. When node 9 generates a packet, it will calculate the sensitive information with the identity of its neighbor and the identity of itself, then embed the sensitive information into the ordinary data. At base Station, the sensitive information will be extracted. Base station will also calculate a value with some items in the received packet. If the calculated value and the extracted information are same, we say the specified neighbor is legal. When there is amalicious node in the network, Node 9 will still calculate the sensitive information with the fake identity of Attacker. But in base station, the validation will find the errors. What is more, for we know the approximate position of Node 9, and Attacker is the neighbor of it, so we can also get the approximate position of Attacker.

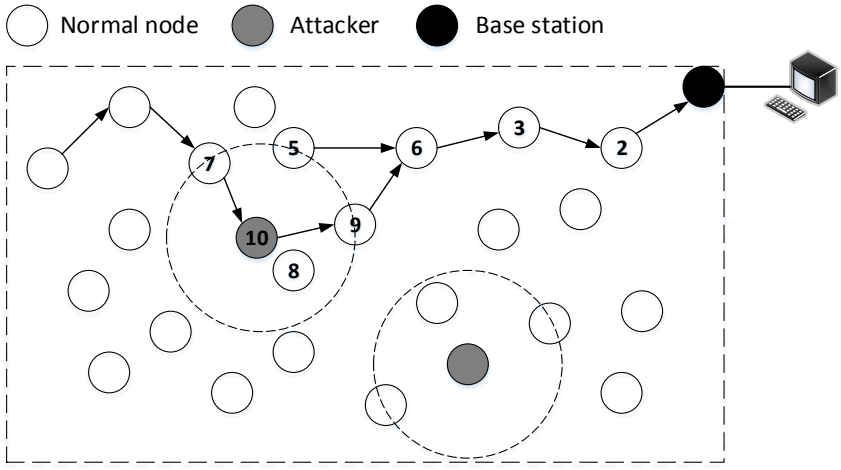


Figure 1. A Topology Built by CTP

In our experiments, the algorithm is loaded into the sending process of TelosB sensors. Experiment results show that, the algorithm can efficiently detect the malicious nodes with fake identities. At the same time, it can ensure the security of sensitive information transmission with a good invisibility, and not affect the transmission and the use of the ordinary data.

The rest of the paper is organized as follows. Section II introduces the related works about CTP, information hiding technology in wireless sensor networks and the basic principles of Bloom Filter. In Section III, we describe our proposed Bloom Filter based embed-

ding/extraction algorithms in detail. The experimental results and performance analysis are presented in Section IV. Finally, Section V includes the conclusion and the future work.

2. Related Works

2.1. Collection Tree Protocol

CTP routing protocol aims to build a tree topology with the collection node as the root [13]. The rest of the sensor nodes use the root advertisements to connect with the collection tree. When a sensor node collects data, it sends the packet up the tree. CTP is an address free protocol, so a node does not send the packet to a particular node but chooses its next hop based on a routing gradient. Each node of the tree forwards its collected data to parent node. In CTP network, route maintains a value called ETX (Expected Transmission Value), which is an integrated value about the link quality of the path. With the ETX value, WSN forms a network gradient. The data from sensor nodes of high gradient is sent to the ones of the low gradient, and finally reaches the sink node. ETX is modified by dynamic calculation periodically. A sensor always chooses the neighbor node with minimum ETX value as the parent node. CTP protocol can be divided into three parts: the link estimator, routing engine and forwarding engine. Link estimator is for estimation between two adjacent nodes in communication quality. Routing engine uses the information provided in the link estimator to choose a node which transmits to the root node with least cost as parent node. Forwarding engine maintenances a send queue which contains local packets and received packets, then it selects an appropriate time to send the head packet of the queue to the parent node.

2.2. Information Hiding in Wireless Sensor Networks

Generally, information hiding technology [8] includes digital watermarking, steganography and fingerprinting.

Digital watermarking technology was used to protect copyright in many applications. In recent years, watermarking technology is also used to protect data security and copyright in WSNs.

J. Feng *et al.* [9] defined sensor relationship with the physical world. The hidden data is embedded by changing the measured values. The extraction of hidden information doesn't need the raw data. However, measured values are not always absolutely correct. The error of measured values will affect the embedding and extraction of watermark.

H.P Guo *et al.* [10] used the watermarking method to verify the integrity of streaming data. The watermark is generated and embedded into the data, which are grouped on synchronization points. The embedded mark can only detect and locate modifications of streaming data.

X. Xiao *et al.* [11] proposed real-time watermarking technique by embedding authorship signature into the sensing data, which imposes error between the expecting distances and ensuring. But embedding information is in some laws, which easily arouses attackers' attention.

Sion *et al.* [14] described a robust watermarking scheme for streaming data proposed for copyright protection. Streams are defined as a continuous sequence of numerical values. The technique identifies key points in the stream called major extremes. A set of major extremes in the group, are identified and selected such that these extremes will survive uniform sampling. The watermark bits are embedded in the major extremes. Thus, trying to destroy the watermark would leave the stream not useful. The watermark can be later extracted and used to proof copyright and ownership of the data stream.

Kamel *et al.* [15] proposed LWC (light-weight chained watermarking) scheme, which simplifies the SGW (sliding group watermark) and avoids several of its drawbacks. LWC uses

chained watermarks; however, it is less complex than SGW. LWC provides significant performance improvement (one to two orders of magnitude in computational overhead over the SGW technique). However, LWC suffers from the same security holes that SGW has.

Therefore, we urgently need more invisible and irregular information hiding scheme to protect the security transmission of sensitive information. At the same time, the secure transmission mechanism should be data-oriented and application-based, and have the ability to make tradeoff between the computation cost and security intensity as well.

2.3. Basic Principle of Bloom Filter

Bloom Filter [16] is a space-efficient randomized data structure. This method classifies elements of a set with a low false positive rate, and saves a large number of storage spaces. Bloom Filter uses a string V with m bits to express the data set $A = \{a_1, a_2, \dots, a_n\}, |A| = n$. Hash function H_i is uniform distribution, where $i \in \{1,2, \dots, k\}$, and $H_i(x) \in \{1,2, \dots, m\}, x \in A$. In the initial state, V contains m bits, all of which is 0. Divide V into groups using hash functions. Each element of A is mapped into some locations of V, and 0 of those locations are changed to 1.

3. Information Hiding Method

3.1. The Modification to Beacon Message of CTP

In our sensor network, every sensor is specified with a unique identity information. So we have to alter the beacon message of CTP to satisfy our need. Figure 2 shows the structure of original beacon message.

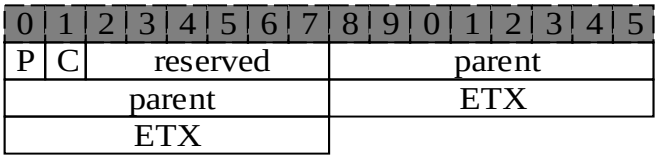


Figure 2. The Beacon Message Structure of Original CTP

The number marked on the top of the figure stands for one bit of space. Here P is routing pull, and its occupation is one bit. The P bit allows nodes to request routing information from other nodes. If a node with a valid route hears a packet with the P bit set, it should transmit a routing frame in the near future. C is for congestion notification and the occupation is one bit. If a node drops a CTP data frame, it must set the C field on the next data frame it transmits. Parent is to mark the node's current parent. ETX indicates the node's current routing metric value.

Once the sensors deployed, they can broadcast beacon message to build the network. So the information in beacon message of local will be recorded by its neighbors. Based on this, we add the identity information sk to sensors, which will be merged into beacon message. The structure of the modified beacon message can be seen in Figure 3 below.

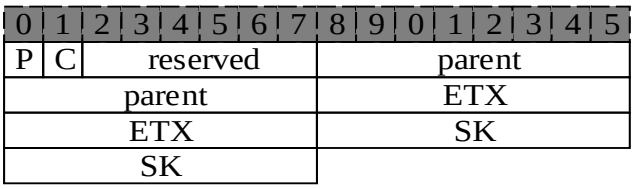


Figure 3. The Modified Beacon Message Structure

Thus whatever the initialization of network or the updating of topology, sensors will get identity information sk from the beacon messages of their neighbors which will be used to generate sensitive information.

3.2. Pre-deployment Phase

Before a sensor is placed, it will specify a sk, which is a unique identity information for it. On the other hand, every sk has a copy stored in the base station, which is corresponded with the node Id of sensor. The existing form is SK, which is,

$SK = \{(node_2, sk_2), (node_3, sk_3), \dots, (node_n, sk_n)\}$,

The base station will get $node_{local}$ and $node_{neighbor}$ from the packet collected by sensor network, and find the corresponding skaccording to the set SK. Then the base station calculates the value in the similar method with which used in the sensor node. If the value and the sensitive information hidden in the packet are the same, we think the sensor with the identity $node_{neighbor}$ is safe.

3.3. Notations and Parameters

In order to facilitate the discussion, the formal definitions of related concepts about the algorithm are given as follows:

Definition 1: The sensor data is defined as S_i,

$S_i = (data_1, data_2, \dots, data_n)$,

Where $i(i = 1,2, \dots, m)$ is the working period, and $data_j(j = 1,2, \dots, n)$ is the item acquired during one working period.

Definition 2: The transmitted packet is denoted as packet,

$packet = (prefix, node_{local}, node_{neighbor}, S, postfix)$,

Where prefix is the prefix of the packet (including some header information), $node_{local}$ is the node ID of the node itself, and $node_{neighbor}$ is the node ID of one of its neighbors, S is the sensor data, postfix is the suffix of the packet (including parity bits, etc.). This packet format is similar with the definition of the MAC in TinyOS.

Definition 3: The random sequence is,

$li[j] = (LI, sk_{local}, sk_{neighbor}, H_i), (j = 1,2,.., LI)$,

Where LI is the string of Bloom Filter, sk_{local} is the identity information of the node itself, $sk_{neighbor}$ is the identity information of the specified neighbor of the node, $H_i(i = 1,2, \dots, k)$ is the hash function of Bloom Filter.

Definition 4: The secure transmission model Q can be denoted as

$Q = (E, EX, LI, C, C')$,

Where E is the embedding function, EX is the extraction function, C is the carrier data set (mostly ordinary data), and $C \subseteq S$, C' is the data set with hidden information. Secure transmission model is the sender to be established.

Rule 1 (Embedding rule): The sender will select several specified elements of S, which is C. LI is embedded into C by modifying LSB of the elements, then the sender will get C', generate and transmit packet' in the monitored sensor network. The embedding process is completed by sender.

Rule 2 (Extraction rule): The extraction process is completed by sink. First, it will get $\text{node}_{\text{local}}$ and $\text{node}_{\text{neighbor}}$ from the received packet', then find the sk of them from the set SK. And $\text{LI}_{\text{onbase}}$ will be calculated in the similar way of the sensor. On the other hand, $\text{LI}_{\text{onnode}}$ will be extracted from LSB of C' in packet'.

3.4. Embedding Algorithm

By Definition 1, it is known that during the working period i, the sensor collects data $S_i = \text{data}_1, \text{data}_2, \ldots, \text{data}_n$, The length of LI equals n.

Algorithm 1. Embedding Algorithm

Input: The carrier data set C, the cursor position pos , and the bloom string LI= n bits, hash function $H_i(i = 1,2,\ldots,k)$

Output: Data set with hidden information C'

Steps:

1) $\boldsymbol{sk_{local}}$ **= call CtpInfo.getSK()**

2) $\boldsymbol{sk_{neighbor}}$ **= routingTable[pos].Info.SK**

3) for i=1 to k

4) if ($\boldsymbol{local_i = H_i(sk_{neighbor})}$**) then li[**$\boldsymbol{local_i}$**] = 1**

5) end for

6) for i=1 to k

7) if ($\boldsymbol{local_i = H_i(sk_{local})}$**) then li[**$\boldsymbol{local_i}$**] = 1**

8) end for

9) for j=1 to n

10) if (li[j] == 1) then $\boldsymbol{C'[last-bit] = 1}$

11) else $\boldsymbol{C'[last-bit] = 0}$

12) end for

13) return$\boldsymbol{C'}$

In the embedding algorithm, it gets sk_{local} from the sensor in step 1. pos is used to mark cursor position in the routing table of the sensor, every time the sensor collects data, pos is increased by 1, the sensor gets $\text{sk}_{\text{neighbor}}$ from routing table in step 2. In step 3-8, getting the random sequence: with sk_{local}, $\text{sk}_{\text{neighbor}}$ and $H_i(i = 1,2,\ldots,k)$, we can get the k locations of LI, and change the corresponding locations of random sequence $li[j](j = 1,2,\ldots,LI)$ to 1; In step 9-12, it is the process of embedding confidential information in sequence. When li[bit] is 1, the least significant bit of $C'[last-bit]$ is 1, otherwise is 0. The process will be recycled until k bits of the confidential information are embedded completely in one packet. The details will be discussed in Section IV.

3.5. Extraction Algorithm

Extracting hidden information is the inverse process of embedding. In the extraction algorithm, we get and check the corresponding locations of random sequence, each of them is 1 or 0. When $C'[last-bit]$ is 1, L[bit] is 1, otherwise L[bit] is 0. The process will be recycled until accomplish in the extraction from one packet.

Algorithm 2. Extraction Algorithm

Input: Data containing hidden information C′
Output: Hidden information L
Steps:
1) …
2) for j=1 to n
3) if (C'_j[last − bit] == 1) then L[bit] = 1
4) else L[bit] = 0
5) end for
6) return L

The base station gets $\text{node}_{\text{local}}$ and $\text{node}_{\text{neighbor}}$ from the received packet′, it will find the corresponding sk_{local} and $\text{sk}_{\text{neighbor}}$ from SK, also it will calculate $\text{LI}_{\text{onbase}}$ in the similar way of the sensor. Then if $\text{LI}_{\text{onbase}}$ equals to $\text{LI}_{\text{onnode}}$, we think the neighbor with ID $\text{node}_{\text{neighbor}}$ is a safe sensor node.

4. Experiment and Performance Analysis

In this section, we evaluate the performance of our proposed secure data transmission scheme. We describe our experiment setup, experiment results and performance analysis in next subsections.

4.1. Experiment Setup

We have tested our scheme using 10 TelosB sensors, each of them includes a mini-USB port for programming and data transfer, an IEEE802.15.4 radio TI CC2420, a low power MCU MSP430 F1611 with 10K RAM, an external flash chip up to 1MB. Figure 4 shows the TelosB sensors used in our experiment. The 10 sensors are densely deployed within an area of 60 square meters, and Node 1 is selected as the sink node.

Figure 4. Sensor Used in our Experiments

In the sensor network, we have run CTP and our scheme consecutively for 3 hours and recorded all the packets received at the sink node. After analyzing the packets, we find that the senders and the sink can effectively ensure the sensitive data transmission, and attackers with fake identities can be detected effectively. At the same time our scheme does not affect the transmission and the use of the ordinary data.

4.2. Experiment Results

1) Detection of fake identity attack

When the sensors in the network have the identity information distributed by the base station, the entire sensor network can work normally. Figure 5 shows a topology of the sensor network in our experiment. When there is no malicious node in the network, each sensor will collect the information around and send it to the base station.

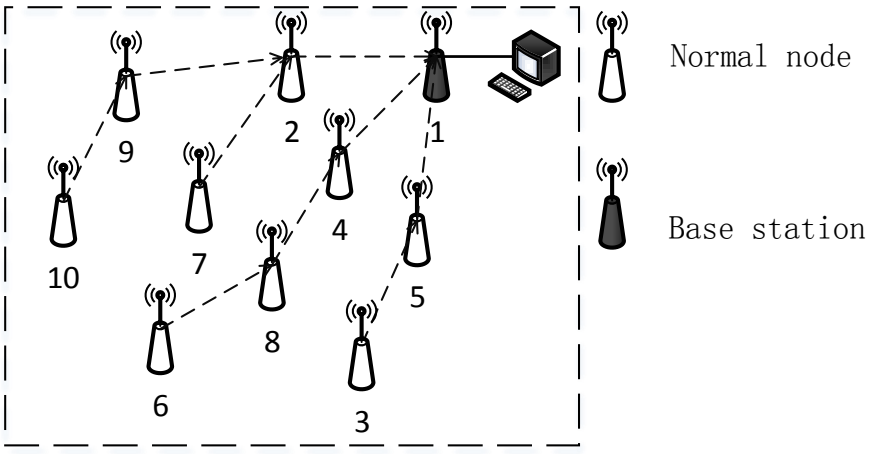


Figure 5. A Topology in our Expriment

However, when exsiting the fake nodes, the base station will send out warnings. In Figure 6, Node 12 is a malicious node, and it does not have the identity information sk given by the base station, but it still wants to access to the sensor network and sends wrong packets. So it has to broadcast beacon messages to inform other nodes nearby, and beacon message is the only way that can make other nodes around know its existence. In figure 6, Node 5 is one of the nodes nearby. And Node 5 will store the identity information of Node 12 in the routing table, although the information is illegal. When Node 5 generates a packet in one period, it will calculate the sensitive information with the identity information from Node 12, and embeds the sensitive information into the packet using the embedding algorithm, then sends the packet to the base station. After the base station gets the packet, it will extract LI_{onnode} from it immediately. Then it will calculate LI_{onbase} by using $node_{local}$and $node_{neighbor}$. But the values of LI_{onnode} and LI_{onbase} are not equal. If the result occurs many times continuously, the base station will regard Node 12 as a tamper, and sends warnings. Then we will find the problem. Because we know the approximate location of Node 5, we can find the malicious node very quickly. After the malicious node is taken, Node 5 will be triggered to update its routing according the characters of CTP, for its route has changed obviously. So the information of Node 12 in its routing table will be erased. In some case, when the identity information of a normal node is used in our scheme, bit errors may happen in the packet of Node 5 during transmission, it looks like a trouble. We will use CRC for the error correction, to ensure the correct extraction of hidden information. In a scarier situation, the CRC does not work when the bit error occurs, the base station will do a statistical assessment to the packets of Node 5 with hidden information of the specified normal node. For the situation is rarely happened that all the packets are illegal. In our experiments, our scheme can effectively detect the malicious nodes with very few mistakes.

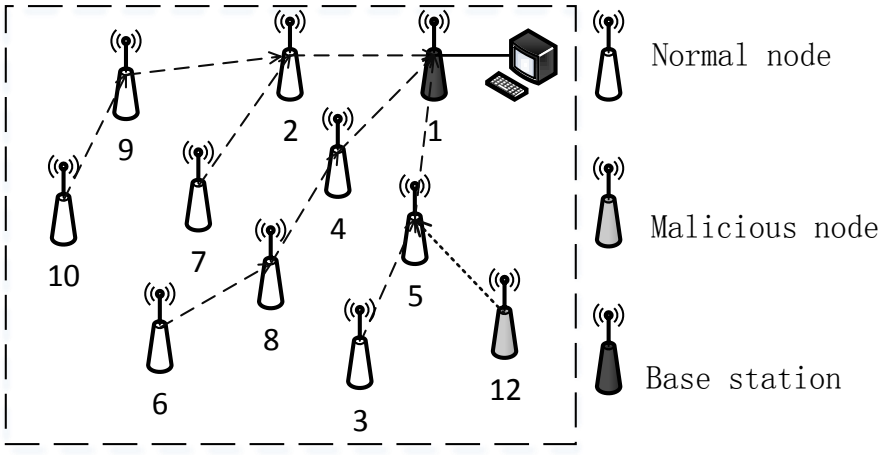


Figure 6. Fake Identity Attack

But in some situations, our scheme will not work. In Figure 7, the attacker does nothing but intercepts the information sent by Node 3. Then the attacker manufactures a node which is a copy of Node 3, and deploys it somewhere else in the sensor network, which is a completely legal sensor in our network, the same ID and the same identity information. But the main purpose of the malicious node is to inject wrong packets to the sensor network. In such case, our base station can also find it. From the figure, we can see that Node 6 is a neighbor of the malicious node, so its information will be recorded by Node 6. When the base station gets packets from Node 6, it will find that Node 3 becomes the neighbor of Node 6, but it should be the neighbor of Node 5 by analyzing the history packets received. Also from the packets send by Node 5, the base station gets that Node 3 is still a neighbor of Node 5. So the base station will send warnings to inform the user. And we know the approximate location of Node 6, so we can find the malicious quickly.

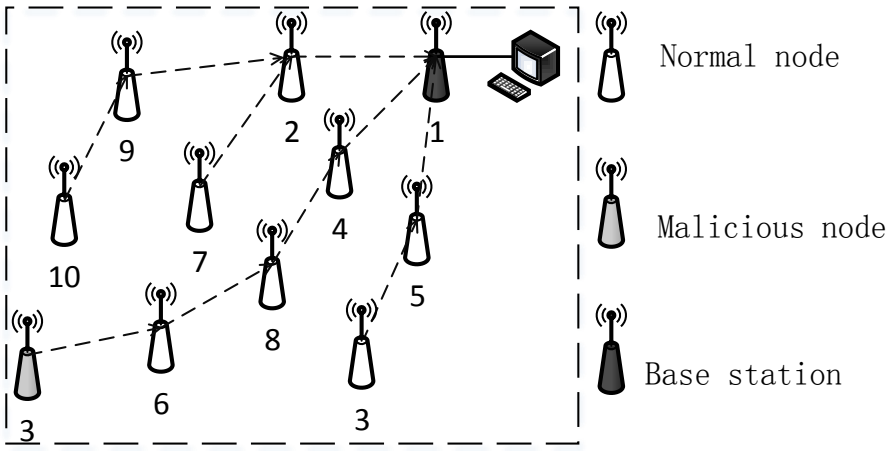


Figure 7. Forged Identity Attack

Another situation is that when a sensor is captured by attacker, our scheme will also not work because its ID and identity information are right. From Figure 8, we can see that Node 3 is captured by attacker, and it seems like a normal node. In such a case, some measures have

been taken. It is considered that a long time will be taked to change a normal node to a malicious node. In our sensor network, each sensor is required to send a packet every five minutes. So when there is a long time the base station cannot receive the packets from Node 3, and then the base station received its packets again after a long time, we think Node 3 is not safe anymore.

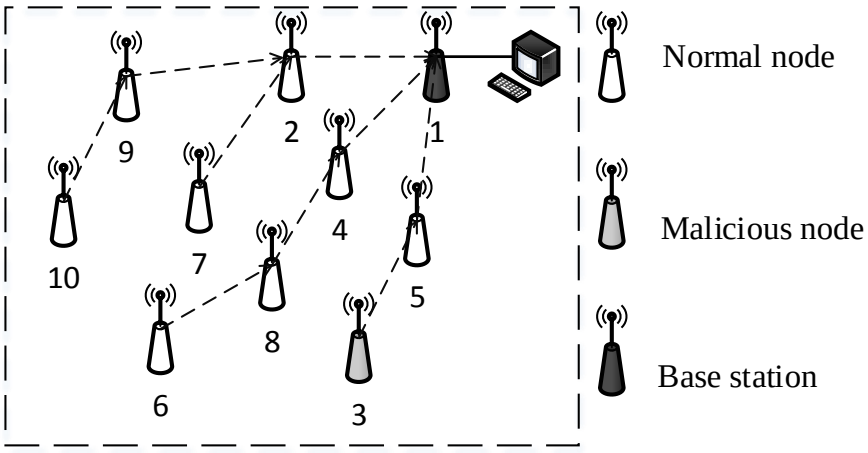


Figure 8. Node Capture Attack

2) Invisibility

In our experiment, the sensor collects 10 data during one working period, so the length of sensitive information is 10 bits. It is uniformly and randomly embedded, so that an attacker cannot easily detect the fixed changes of data. If attackers believe that there is some information hidden in the packets, the probability of obtaining k locations is $\prod_{i=0}^{k-1} \frac{1}{n-i}$ by violence attack. In addition, the false positive of Bloom Filter also increases the difficulty of attack. In one word, the algorithm is dynamic, random and covert and increases the difficulty of monitoring and attacking. It can effectively prohibit the attacker getting sensitive information. On the other hand, numerical data are maintained changing in the range of ± 1 according to LSB. The following figure reflects the changes of the embedded data. In one period, the sensor will collect 10 data, and we choose one item of them to state the issue. The value 122 is a normal data collected by a sensor when it comes to embed the sensitive information, if the correspondingsensitive information bit is 1, the value will be modified to 123. If the bit is 0, the value will remain unchanged. What is more, the sensor calculates the sensitive information every time, it will choose identity information of different neighbors at this level and it also promotes the randomness. For the general application, changes within a small range are usually acceptable, and it is also difficult to be detected.

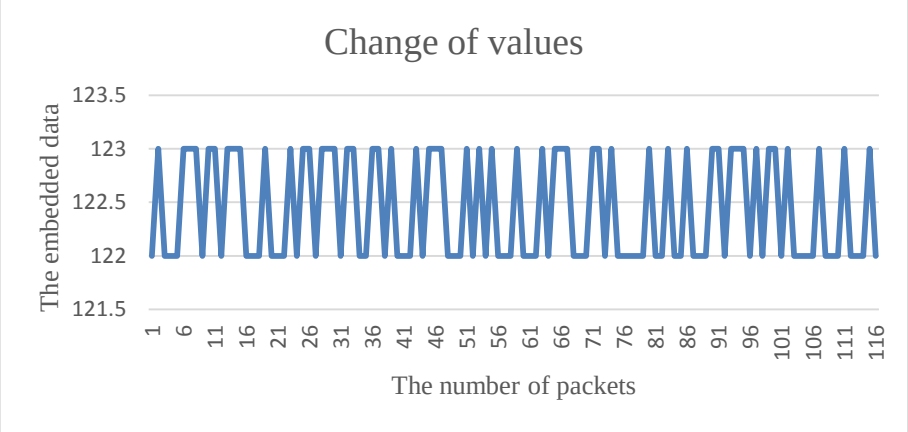


Figure 9. Changes of Value after Embedding

3) Robustness

When we embed the sensitive information into the ordinary data, our scheme satisfies all the required security properties of digital watermarking.Figure 10 reflects that when there is no tamper attack, the rate of correct exaction can be 100%, which means our information hiding technology is stable. If there is bit error, CRC can be used for the error correction, to ensure the correct extraction of sensitive information.

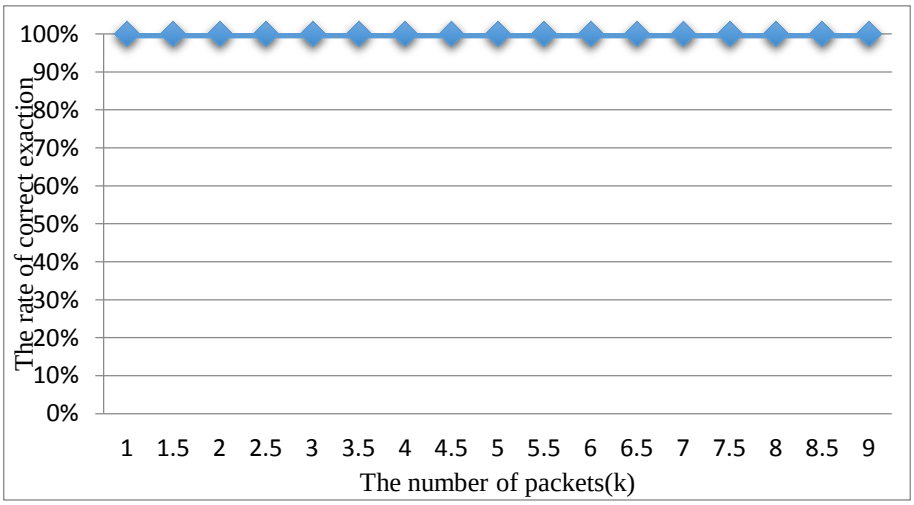


Figure 10. The Rate of Correct Exaction when there is No Attack

4) Energy consumptions

In our scheme, every sensor is loaded with a unique identity information. So we add 16 bits in the beacon message of CTP. Because the formation of routing relies on it, which means it will take more energy to broadcast the extra bits. On the other hand, data embedding will also consume energy. But in our experiments, the communication radius of a sensor is about 100 meters, and the distance of one hop is within the range of 50 meters. According to $E = kd^n (2 < n < 4)$, the energy consumed by beacon broadcasting is very few. And in our ex-

periment, 3 hours later, the voltage of each sensor is still around 2.9992V or 2.9985V, which is a very small range. So we think the energy consumption is acceptable.

5. Conclusions and Future Works

This paper presents a novel algorithm to deal with malicious nodeswith fake identities and ensure the secure transmission of hidden information. In this scheme, we change the beacon message and routing table of CTP in order to satisfy the need of transmitting and storing identity information of sensors. Sensitive information will be uniformly and randomly embedded into the hidden locations of the ordinary data by using Bloom Filter. Experimental results show that our algorithm can achieve the covert transmission of sensitive information, and it has the robustness to some attacks of adversaries in sensor networks. By analyzing the packets received, we can find the malicious nodes in our network. The algorithm has a little additional costs, and the influence will be acceptable in the life time of the entire network for embedding and transmitting sensitive information. Although this scheme is able to hide sensitive information effectively and avoid adversaries' attention, it cannot completely resist various attacks. Therefore, we should improve its robustness to against various attacks in the future.

Acknowledgements

This work is supported by the NSFC (61232016, 61173141, 61173142, 61173136, 61103215, 61070196, 61070195, and 61073191), National Basic Research Program 973 (2011CB311808), 2011GK2009, GYHY201206033, 201301030, 2013DFG12860, PAPD fund and NUIST Research Fund.

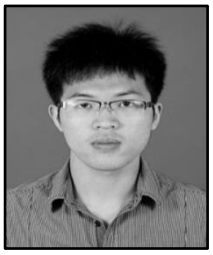
References

[1] N. Cao, Q. Wang, K. Ren and W. Lou, "Distributed Storage Coding for Flexible and Efficient Data Dissemination and Retrieval in Wireless Sensor Networks", IEEE International Conference on Communications, **(2010)** May 23-27, Cape Town.
[2] K. Ibrahim and H. Juma, "A Lightweight Data Integrity Scheme for Sensor Networks", Sensors, vol, 11, no. 4, **(2011)**, pp. 4118-4136.
[3] A. S. K. Pathan, H. Lee and C. Hong, "Security in Wireless Sensor Networks: Issues and Challenges", Advanced Communication Technology, the 8th International Conference, **(2006)** February 20-22, pp.1043-1048: Phoenix Park, Korea.
[4] P. Bartosz, D. Song and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks", Journal of Computer Security, vol. 15, no. 1, **(2007)**, pp. 69-102.
[5] X. Ren, and H. Yu, "Security Mechanisms for Wireless Sensor Networks", International Journal of Computer Science and Network Security, vol. 16, no. 3, **(2006)**, pp. 155-161.
[6] A. Durresi, V. Paruchuri, R. Kannan and S. S. Lyengar, "Data Integrity Protocol for Sensor Networks", International Journal of Distributed Sensor Networks, vol. 1, no. 2, **(2005)**, pp. 205-214.
[7] R. Venugopalan, P. Ganesan, P. Peddabachagari, A. Dean, F. Mueller, and Sichitiu, "Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis", In Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, **(2003)** October 30-31, pp. 188-19: New York, USA.
[8] J. Y. Halpern and R. O. Kevin, "Anonymity and Information Hiding in Multi-agent Systems", Journal of Computer Security, vol. 13, no. 3, **(2005)**, pp. 483-514.
[9] J. Fexng and M. Potkonjak, "Real-time Watermarking Techniques for Sensor Networks", Security and Watermarking of Multimedia Contents, vol. 5020, **(2003)**, pp. 391-402.
[10] H. Guo, Y. Li, "Chaining Watermarks for Detecting Malicious Modifications to Streaming Data", Information Sciences, vol. 177, no. 1, **(2007)**, pp. 281-298.
[11] X. Xiao, X. Sun, L. Yang and M. Chen, "Secure Data Transmission of Wireless Sensor Network Based on Information Hiding", MobiQuitous 2007. Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, **(2007)**August 6-10, pp. 1-6: Philadelphia.

[12] H. Wang, D. Peng, W. Wang, H. Sharif, and H. Chen, "Energy-Aware Adaptive Watermarking for Real-Time Image Delivery in Wireless Sensor Networks", IEEE International Conference on Communications, **(2008)** May 19-23, pp. 1479-1483: Beijing, China.
[13] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss and P. Levis, "Collection Tree Protocol", ACM Conference on Embedded Networked Sensor Systems - SenSys, **(2009)** November 1-3, pp. 1-14, New York, USA.
[14] R. Sion, M. Atallah and S. Prabhakar, "Resilient Rights Protection for Sensor Streams", In Proceedings of 30th International Conference on Very Large Data Bases, **(2004)** September 24-26, pp. 732-743, Toronto, Canada.
[15] I. Kamel and H. Guma, "Simplified Watermarking Scheme for Sensor Networks". International Journal of Internet Protocol Technology, vol. 5, no. 12, **(2010)**, pp. 101-111.
[16] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors", Communications of the ACM, vol. 13, no. 7, **(1970)**, pp. 422-426.

Authors

Baowei Wang, he received his B.S. and Ph.D. degrees in Computer Science from Hunan University in 2005 and 2011, respectively. He is currently working as a lecturer in School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include steganography, wireless networks and securing ad hoc networks.



Hongwei Qian, he received his B.S. degree in Computer Science from Nanjing University of Information Science and Technology, China in 2011. Currently he is studying for his M.S degree in Meteorological Information Security at the same university. His research interests include wireless networks and network security.

Xingming Sun, he is a professor in the School of Computer and Software, Nanjing University of Information Science and Technology, China from 2011. He received the B.S.degree in Mathematical Science from Hunan Normal University and M.S. degree in Mathematical Science from Dalian University of Technology in 1984 and 1988, respectively. Then, he received the Ph.D. degree in Computer Engineering from Fudan University in 2001. His research interests include information security, network security, cryptography and ubiquitous computing security.

JianShen, he received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2007 and the M.E. and Ph.D. degrees in Computer Science from Chosun University, Gwangju,Korea, in 2009 and 2012, respectively. Since late 2012, he has been a Professor in the College of Computer and Software at Nanjing University of Information Science & Technology, Nanjing, China. His research interests include computer networking, security systems, mobile computing and networking, ad hoc networks and systems, and ubiquitous sensor networks.

Xiaoyu Xie, he received the B.E. degree from the Naval University of engineering, Wuhan, China, 2007, he has been a engineer in information and technical support at Wuhan Meteorological Service, Wuhan, China. His research interests include computer network, meteorological equipment.