An Efficient Compromised Nodes Detection System in Wireless Sensor Networks

Xiaolong Xu[1], Zhonghe Gao[2], Lijuan Han[1]
(Corresponding author: Xiaolong Xu)
Experiment Teaching Center, Qufu Normal University[1]
Rizhao Shandong 276826, China
(Email: xiaolongxu@foxmail.com)
Institute of Software, Qufu Normal University[2]
Qufu Shandong 273165, China
(Received Jan. 26, 2018; revised and accepted Apr. 28, 2018)

Abstract

Wireless sensor networks have limited resources and are deployed in an open environment, this makes it vulnerable to attacks. The CND method we proposed can accurately detect the compromised nodes in wireless sensor networks. Experimental results show that the CND system has the following advantages: The detection rate and false positive rate are better than the existing compromised node detection methods; It is not vulnerable to slander attacks; It can run in most wireless sensor networks and can automatically adjust the detection behavior according to the network transmission; It requires only small memory and low communication overhead, so it can be applied to large-scale networks.

Keywords: Attack Detection; CND; Compromised Node; Wireless Sensor Network

1 Introduction

Sensor nodes are cheap and autonomous, this extends wireless sensor networks to several applications, including environmental monitoring, medical care, smart home, traffic control, and so on. However, the application of wireless sensor networks has been extended to many security fields, and information security has become an important aspect of people's concern. In hostile environments, it is impossible to trust reports from wireless sensor networks without information security.

However, sensor nodes only have limited resources, such as the limited computing power, memory and battery life, and are usually deployed in an open environment, so they are vulnerable to attacks, and attackers can control some nodes [14]. If not detected, a compromised node is considered a network's authorized participant, which can use its own authority to launch an internal attack.

Therefore, security oriented wireless sensor networks must take measures to prevent node compromise. Generally, security policy can be divided into prevention, detection and recovery. Because of the small size and low cost of micro sensor nodes, limited resource limits the effectiveness of the defense mechanisms [15]. Attackers can use more powerful machines, such as laptops, to capture nodes. So defensive measures can only delay an attacker's attack.

Attacks on wireless sensor networks can be divided into external attacks and internal attacks. The external attacker is located outside the wireless sensor network, and the internal attacker is the authorized user of the wireless sensor network. Both external attackers and internal attackers can capture sensor nodes and make them compromised nodes. Malicious codes are running on compromised nodes, so they become nodes controlled by attackers, which seriously threaten the security of wireless sensor networks. Compromised nodes detection is of great importance for ensuring the security of wireless sensor networks [10].

Detection mechanism is an active measure to prevent node compromise. Once the compromised node is detected, appropriate measures are taken to reduce the loss caused by compromise. At present, there are many kinds of compromised node detection methods, but they have various disadvantages. In addition, most of the detection methods are for specific situations, and they do not perform well in other cases. For example, most of the detection systems do not consider lossy environments, and packet loss is more common in wireless sensor networks. The packet loss rate of 20% reduces the detection rate by more than 50% and leads to false positive rates of over 90% [5].

The intrusion detection system CND (Compromised Nodes Detection) is proposed to identify the compromised nodes in wireless sensor networks. This detection method has the following advantages:

Accuracy: CND can detect the compromised nodes in wireless sensor networks timely and accurately. Specifically, it requires a high detection rate and low false alarm rate and short detection time. High detection rate means that the vast majority of compromise behaviors can be detected. Low false positive rate means that most reports of compromised nodes are accurate, so you can rest assured that such nodes should be taken corresponding measures. Finally, shorter detection time limits the amount of malicious activities that a compromised node can perform before being detected.

Flexibility: CND does not change for a particular application or deployment because it limits its scope of application. It considers the underlying network as little as possible and can be used in most situations to detect compromise behaviors.

Robustness: Compromised nodes may try to damage detection system through malicious behaviors, such as slander attack. They will send false information so that the legitimate node is mistaken for a compromised node. CND must be able to prevent such malicious behaviors. Even with full knowledge of CND, attackers can't use it to control the rest of the network.

Extendibility: Because micro sensor nodes only have limited resources, some high cost applications will interfere with other applications and reduce the lifetime of sensor nodes [12]. CND has very low overhead, so it has only a minor influence on other applications deployed in the network.

The main goal of CND is to provide a system to identify compromised nodes accurately, so as to improve the overall security of wireless sensor networks. CND uses a lightweight distributed architecture that can be deployed in resource limited devices, such as wireless sensor nodes. It has little influence on other applications and network lifetime. Through many experiments, we find that CND has more accurate detection ability than other similar systems, and can be extended to tens of thousands of nodes.

The rest of this paper is structured as follows: The first chapter reviews the previous research of compromised nodes detection in wireless sensor networks. The second section discusses the proposed system and threat model. From Sections 3 to 6, the design, implementation and evaluation of CND are introduced respectively. Section 7 is the conclusion and future work.

2 Literature Review

Most detection methods of wireless sensor networks mainly focus on some specific attacks, such as node replication attack, wormhole attack, sybil attack, etc [9]. Although they may detect compromised nodes indirectly, attackers can escape detection by avoiding target attacks.

The traditional method of detecting compromised nodes is authentication. The authentication method is to check the changes of node memory to find out whether the modified code is running. The advantage of this approach is the ability to detect compromised nodes that do not perform destructive activities. A variety of software based authentication techniques have been proposed for wireless sensor networks [3], but software based security authentication has not been implemented in wireless sensor networks yet.

Other programmes focus on monitoring suspicious communication behaviors. Suspicious behavior can be confirmed by anomaly detection or rule based detection. Anomaly detection establishes a baseline of normal behaviors and considers a behavior abnormal when detected beyond baseline. For example, intrusion detection system proposed by Onat and Miri mainly monitors two features - packet arrival rate and received power [13]. The detection nodes continuously monitor these two features from adjacent nodes and are considered abnormal if new data is found to deviate from the established baseline. Malicious behaviors that can cause changes in these two features will be detected, such as replay attacks. Rule based detection judges a behavior as malicious when the behavior is found to consistent with the rules set earlier. For example, the COOL system is an intrusion detection system that detects the compromised nodes using the relationship between incoming and outgoing messages [16]. The COOL system is based on the idea that the vast majority of outgoing messages should be forwarded to incoming messages. When a node sends more information than it receives and reaches a threshold, it is considered a compromised node.

Compared with the existing methods, the proposed method is more flexible, robust and scalable. CND can accurately detect compromised nodes in the presence of packet loss, without being affected by other applications running on the sensor nodes. It is very effective in combating large-scale slander attacks. In this attack, the compromised nodes hinder the detection process. In addition, it has the advantage of low overhead. This allows it to be deployed in a wireless sensor network with thousands of nodes without affecting other applications deployed, without significantly shortening the lifetime of the network.

3 System and Threat Model

A CND system is designed based on the following common features of wireless sensor networks and compromised nodes.

1) The sensor nodes are densely deployed in the network, so that the sensor nodes have overlapping perception range. Thus, an event may be detected by multiple nodes at the same time. Because of range overlap, one sensor node can monitor the behavior of its neighbors.

2) Sensor nodes have limited energy, calculation ability, and communication capability. For example, the Mica2 micro sensor uses an Atmel microprocessor with a main frequency of 4 MHz and a word length of 8 bits, it is equipped with 128 KB instruction memory and 4 KB RAM.

3) A routing protocol that forwards messages between the base station and the node is required.

4) A base station is a higher order device, such as a computer placed in a secure location.

5) The sensor node has a unique identifier that enables the base station to know which node corresponds to the reported compromise behavior.

6) All messages have time stamps.

7) Attackers can capture nodes either by physical capture or by means of wireless communication channels. Once a node is compromised, all the information, including the key, is acquired by the attacker.

8) Although compromised nodes can perform any number of attacks to reduce the security of network, this paper focuses on compromised nodes that perform malicious behaviors, such as forges and tampers with data.

CND can make use of these characteristics to achieve accurate identification of compromised nodes, and only a small amount of overhead is needed.

4 System Architecture

When designing CND, you must determine whether to use a distributed, centralized, or hybrid architecture. Building a pure distributed intrusion detection system is very challenging because the limited resources of sensor nodes restrict the use of complex algorithms. For example, traditional security protocols, such as modulo operations used by RSA, run more difficult on 8 bit node processors. Complex computations can be distributed over a number of sensor nodes, but nodes that engage in critical operations can become compromised nodes, resulting in spurious results. In contrast, centralized solutions do not have these problems because the base station has more resources and is more secure. However, the data received by the base station from compromised nodes may be false. Therefore, the network needs to have some degree of additional functions to detect false data from the nodes.

Thus, CND takes a hybrid approach, as shown in Figure 1. The system consists of two parts: a distributed system running on each node in the network and a centralized system running on the base station.

Distributed component: Copies of this component run on each sensor node and run concurrently with applications, routing protocols, and so on. Each copy is responsible for detecting possible compromise behavior among adjacent nodes and reporting to the base station. This detection relies on adjacent node monitoring, and each node records and analyzes the behavior of its neighboring nodes. Because of the broadcast characteristics of wireless sensor networks, this does not cause excessive communication overhead.

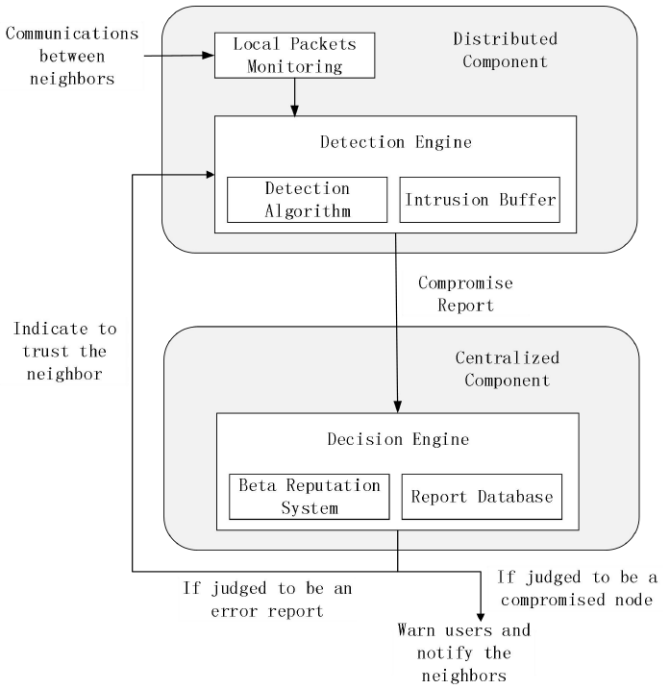


Figure 1: The framework of CND

Centralized component: A base station is a higher level device, so CND uses it to perform complex analysis to determine whether a reported compromise is correct. The base station collects data from the entire network, which is what the sensor nodes cannot do with their own local views and limited resources.

After the network deployment, there is an initial setup phase. During this phase, nodes establish their neighbor lists, routes to base stations, and so on. The network is safe for some time after the initial deployment, and this phase does not introduce any vulnerabilities because the attacker cannot immediately capture a node after the network has just been deployed. If this requirement cannot be reached, then the information must be preprogrammed to each node before the network is deployed.

4.1 Distributed Component

Each sensor node has a distributed component running on it that will record data from neighboring nodes and establish baselines based on these records. The baseline indicates the normal behavior of the nodes, and the behavior that deviates from the baseline will be considered

an abnormal behavior. If a neighbor node continues to perform an abnormal behavior, it will be identified as a compromised node and reported to the base station.

Considering instantaneous errors, such as collisions or other unmalicious behaviors, CND is flexible in determining a node as a compromised node and can tolerate certain abnormal behavior. When judging whether a neighbor node is abnormal, there is no cooperation between nodes. This independent decision process implies that the compromised nodes can not affect the perspective of legitimate neighbor nodes.

4.1.1 Monitoring Features

The first step in designing any security system based on detection is to select the system features to be monitored. To support most wireless sensor networks, CND monitors only common features of wireless sensor networks.

1) Sensor reading: By monitoring sensor readings, attacks attempting to distort the collected information can be detected [4].

2) Received power: In a static network, the received power should remain constant. Fluctuations may be caused by changes in the location of communication hardware or corresponding nodes.

3) Sending rate: Most applications read sensor readings and periodically send them. Routing packets are also sent periodically. Therefore, the rate at which packets are sent by nodes should follow a consistent pattern. Most attacks, such as selective forwarding, sybil attack, replay attack, etc., can cause metric deviation. In addition, a sudden idle period may be caused by opponent's rewriting node program.

4) Receiving rate: The ratio of incoming and outgoing packets should be constant, because the outgoing packets can only be those routed or generated by nodes. A neighbor node whose receiving rate has changed, but its sending rate does not change, such a node may be a compromised node. It should be noted that, regardless of whether the data is encrypted, the header of a packet is usually visible to all nodes.

Because most wireless sensor networks have these characteristics, CND has a wide range of applicability. However, these features may not be appropriate for two scenarios: (1) Packets can only be decrypted by base stations; (2) Applications rarely communicate with base stations.

The first scenario will appear when the confidentiality of the information is very important. Since compromised nodes cannot be detected immediately and blocked, some sent messages may be tapped by compromised nodes. Therefore, packets can be encrypted and only base stations can decrypt them. Under such conditions, the number of monitored neighbors can be increased to make up for defects that cannot monitor sensor readings, thus enabling CND to achieve appropriate performance by occupying a little more memory.

The second scenario is caused by applications that are not periodically communicated. For example, wireless sensor networks in a demilitarization zone send messages only when an attack is detected, and they do not communicate in a secure environment. Due to insufficient monitoring information, the baseline cannot be established for most features. CND compensates by making the node send its unique identifier at a certain speed. Long silence will cause the overcome nodes cannot be found, therefore a certain amount of communication overhead is needed. This behavior pattern is used only when the amount of communication in the application is small.

4.1.2 Detection Algorithm

There are two kinds of algorithms for detecting abnormal behaviors: anomaly detection and rule based detection. They all use records of monitoring system characteristics. The anomaly detection algorithm uses the existing record to establish the baseline, and any new record that deviates from the baseline to a certain extent is considered to be an abnormal behavior. On the contrary, rule based detection should establish a specific standard. For example, any two packets have the same header means a replay attack occurred. In CND system, the main attention is paid to anomaly detection algorithms to meet the requirements of CND for flexibility. Rule based algorithms aim at special situations, and the rules must be updated for each new situation.

The distributed component of CND can be divided into five algorithms for detecting attack behaviors: The first four algorithms are anomaly detection algorithm, which uses network features such as sensor reading, received power, sending rate and receiving rate; The fifth algorithm is a rule based detection algorithm.

For rule based algorithms, if a node detects a new neighbor that conforms to the characteristics of the previously predefined rule base, it is considered that the new neighbor is a compromised node.

These rules can prevent compromised nodes and external attackers masquerading as normal nodes without being discovered, and Figure 2 illustrates this nature. Suppose that node A is compromised and want to impersonate another node, if node D does not detect new neighbors, it cannot impersonate B or C; if node B and C do not detect new neighbors, it cannot impersonate D; if node B, C, D do not detect new neighbors, it cannot impersonate any other node. Therefore, if there are enough neighbors to monitor each other, any attack and impersonation can be detected.

All anomaly detection algorithms follow a similar approach. Each node sets two buffers for each monitored neighbor: a packet buffer and an abnormal behavior buffer. All anomaly detection algorithms share the buffer and use the sliding window mechanism. It stores the last

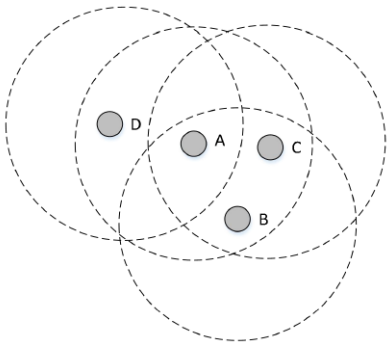


Figure 2: Node A must be detected by neighbor nodes if it wants to impersonate other nodes

N packets or reports of the corresponding neighbors. The data stored in the packet buffer is used to compute the baseline of the neighbors. The new packet is compared with the baseline, and any packet that deviates from the baseline beyond a certain threshold is considered to be abnormal. The abnormal packet indicates the intrusion behavior, and causes the detection node to produce abnormal behavior report. All the reports are added to the abnormal behavior buffer. When the cumulative report number in the abnormal behavior buffer exceeds the threshold, the node will report the corresponding neighbor to the base station as a compromised node.

An overview of the algorithm that uses the received power is shown in Figure 3, and the corresponding equation is:

$$power_{new} - power_{max} > T, \quad \text{if } power_{new} > power_{max}$$
$$power_{min} - power_{new} > T, \quad \text{if } power_{new} < power_{min}$$

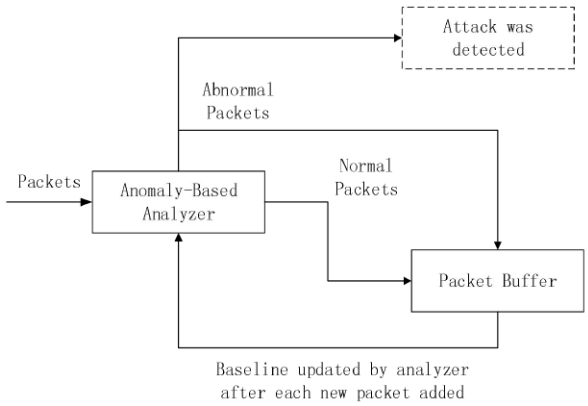


Figure 3: Overview of the detection algorithm using received power and sensor readings

This algorithm calculates the maximum and minimum values of packet received power in the packet buffer. If the received power of a new packet is lower than the minimum value of T or higher than the maximum value of T, it is considered to be abnormal. Abnormal data packets are added to the packet buffer so that anomalies caused by environmental changes can be taken into account when calculating baselines in the future.

The sensor reading algorithm is almost the same as the algorithm using the received power, and the only difference is the use of sensor readings from nodes and neighbor nodes instead of the received power.

Figure 4 shows an overview of the algorithm that uses the sending rate. It calculates two rates: The sending rate of the last N_2 packets $rate_{N_2}$ and the sending rate of the last N packets $rate_N (N > N_2)$. If the ratio of these two rates is higher than the threshold K, the corresponding neighbor nodes are considered to be compromised nodes.

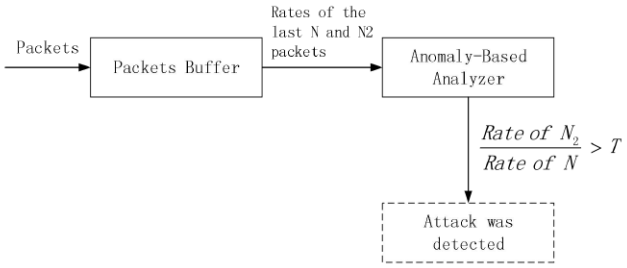


Figure 4: Overview of the detection algorithm using sending rate and receiving rate

The algorithm that uses receiving rate differs only in two ways. First, instead of calculating packets sent by neighbors, the data packets received by neighbors are calculated. Second, the rate is replaced by the ratio of the sending rate and receiving rate, that is, $rate_{N_2}$ becomes $rate_{sent_{N_2}}/rate_{rec_{N_2}}$, and $rate_N$ becomes $rate_{sent_N}/rate_{rec_N}$.

All illegal behaviors detected by anomaly detection algorithm are stored in a shared illegal behavior buffer. Each reported illegal behavior is assigned a weight based on the detection time t_{stamp} and the current time $t_{current}$. When a neighbor's illegal behavior is detected, the weight of its illegal behavior is calculated:

$$\sum_{M}(t_{current} - t_{stamp}) + 0.3\sum_{m}(t - current - t_{stamp}). \quad (1)$$

Where M represents all detected illegal behaviors of the same type, and m represents all other types of illegal behavior. When the result of Equation (1) exceeds the threshold T_M, the corresponding neighbors are considered to be compromised. After thousands of simulations, the weight 0.3, the optimal value of the threshold and other parameters of the equation can be determined.

Once a node determines that a neighbor A is compromised, it sends three reports about this node to the base station. Each of these reports has three domains, which are reporter, reported node, and illegal behavior type. Since then, the reporter will continue to record information from node A, but will no longer detect abnormal behavior unless instructed by the base station.

4.2 Centralized Component

A centralized component runs on a base station to determine whether a reported node is really compromised based on data from other nodes. If the reported node is a compromised node, the base station will notify the user and execute the recovery process, such as ignoring all the messages of the node. If the reported node is not compromised, the base station will notify the reporter to treat it as a non compromised node and continue to monitor it.

Users will receive notifications for all new neighbor reports. If an actual node is added to the network, the user can notify the network that the node is not malicious. For other cases, the base station will process data based on reports from other nodes. In order to determine whether a reported node is compromised, CND uses the beta reputation system [12]. Research shows that Beta reputation system can accurately detect illegal behavior and reduce false positives rate based on a large number of reports. Because the system takes historical factors into account, in order to successfully hide a compromised node A, the average 72% of the neighbors of node A need to become compromised nodes. This is better than other programs (such as Majority Voting) of 33% 50% [7]. Beta reputation system uses probability density function and multi source feedback to determine reputation rating. For this paper, reputation rating is to judge whether a node beyond the threshold is a compromised node.

In Beta reputation system, the probability is ρ, each reported event is given two parameters α and β of beta distribution. $f(\rho|\alpha,\beta)$ can be represented by Γ function:

$$f(\rho|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)}\rho^{\alpha-1}(1-\rho)^{\beta-1}, \quad 0 \le \rho \le 1, \alpha > 0, \beta > 0.$$

The parameter α and β denote the weighted sum of all previous reports of the reported nodes and the number of compromised nodes within two hops of the reported node. This allows the network topology and past reports to influence the final decision on whether a reported node is a compromised node. The initial baseline value is determined during installation.

The base station knows the neighbor information of each node, and this information is collected during the network installation. If there are more than one neighbor reporting that A is a compromised node, then there is a higher likelihood that it is right. The longer the history of the report, the more compromised nodes there may be.

On the other hand, if the reported node is unlikely to be a compromised node, then the report node B may be a compromised node and initiate a slander attack on node A. In this case, the base station will notify the other nodes that node B is a compromised node, and alerts the user, and starts the recovery program.

This method can prevent attackers from using CND to attack the network without being detected. If a compromised node poses as a base station, the nodes near the base station on the routing path will detect messages from the wrong direction and alert the base station. Therefore, once an attacker is posing as a base station, it will be detected immediately.

CND is not vulnerable to slander attacks. As mentioned earlier, masquerading as other nodes will be detected by neighbor nodes, and the nodes do not affect each other. Suppose a compromised node C wants to slander its neighbor node D, because the base station knows the neighbor of node C, so reporting a non neighbor node can also lead to detection. The only possible slander attack is that node C affects base stations by sending reports on compromised neighbors. If there is no support report from the neighbor of node D, the base station will not consider node D to be a compromised node. Slander attack only leads to node C being considered a compromised node.

5 System Implementation

This paper uses TinyOS operating system to implement CND [1]. There are two key issues that might be applicable to other implementations of CND:

First, each node has an illegal behavior buffer to store the illegal behaviors of neighbor nodes. The format and size of the buffer are related to the specific implementation. For example, it depends on the required accuracy and the performance of wireless sensor networks. The report in the buffer must contain the following domains: alarm time, illegal behavior type and source.

Second, monitoring all neighbors makes the buffer require a higher memory overhead. Memory overhead is exponentially increased by the number of direct neighbors, so the overhead for high-density networks is very large. To solve this problem, CND nodes can select a subset of neighbor nodes to monitor, and the selection can be random or in accordance with other protocols. For example, in the random pairwise key distribution protocol [8], some keys are generated before deployment, and each node is assigned a random key. After deployment, it is possible that two neighbor nodes have compatible keys and can communicate with each other. The number of neighbors monitored by each node is controlled by the density of the network, so that each neighbor can communicate with it. This paper will show in the following chapters that when the number of monitored neighbors reaches a certain value, the performance of CND will reach its maximum. So, in dense networks, there is no need to monitor all neighbors.

6 System Performance

In order to analyze the performance of CND, a series of experiments were carried out using SenSec [17]. SenSec is an evaluation tool that enables people to imitate and analyze various attacks in wireless sensor networks. The

validity of CND in different parameters is quantitatively analyzed.

In this paper, the standard performance indicators of the detection system are as follows:

1) Detection rate. This indicator is a percentage of the actual compromised behaviors detected by the system. However, even if the detection rate is 100%, the accuracy of the system can not be determined without considering the false positives.

2) False positive rate. Legitimate nodes can be erroneously reported as compromised nodes, and these reports are called false positives. The detection rate is not inversely proportional to the false positive rate. The system with high false positive rate is inaccurate, because most of the reported compromised behaviors are false.

3) Detection time. Before determining whether a node is compromised, the detection mechanism takes time to process the collected data. Detection time refers to the time that a compromised node keeps the state of being not detected.

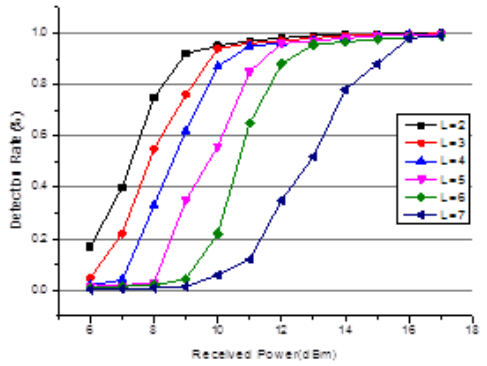
The performance of the distributed component and the overall performance of the CND will be discussed below.

6.1 Performance of Distributed Component

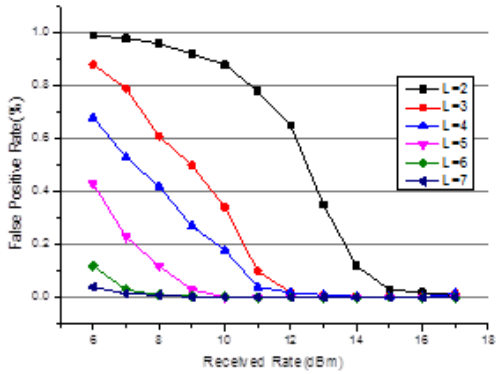
When modeling the compromised nodes in network, a gradient based model proposed by Chen *et al.* is used [2]. The model is based on the perspective of the spatial locality of the compromised node. For example, if a node is close to a compromised node, it will be more likely to become a compromised node. Therefore, the probability of a node being conquered forms a gradient, the closer to the compromised node, the more likely it will be conquered.

The experimental network topology consists of 100 analog nodes randomly deployed in a $100m \times 100m$ area. The node has a wireless transmission device with a transmission power of 5 dBm, and runs a universal sensor application, reads the sensor readings every second, and routes them to a base station at any edge of the network. The tree routing protocol and CSMA protocol are used in the experiment. First, the system is set up. At a random time after the setup stage, a random node in the analog wireless sensor network is conquered every 10 simulated minutes, and a series of attacks on the network are launched. Attacks initiated by compromised node are provided by SenSec, such as replay attacks, witch attacks, wormhole attacks, pulse delays, selective forwarding, and so on. In the simulation, each node runs a real TinyOS application with a sensor readings every 0.1 s. Each experiment includes 50 runs, and each run lasts for 1 simulated hours.

Figure 5(a) and 5(b) show the experimental results, which can be used to evaluate the performance of different received power detection algorithms. For the original packets, the constant 5 dBm transmission power is used in this paper, while the received power is simulated according to physical topology, and then the level of transmission power increases gradually.



(a) Detection rate



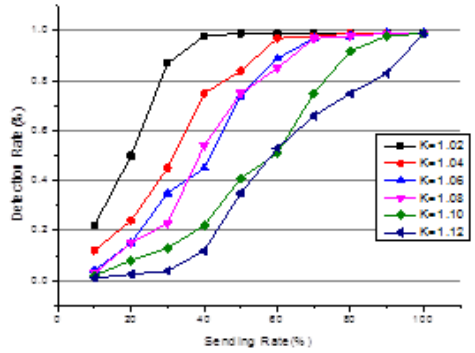
(b) False positive rate

Figure 5: Performance of detection algorithm based on received power change

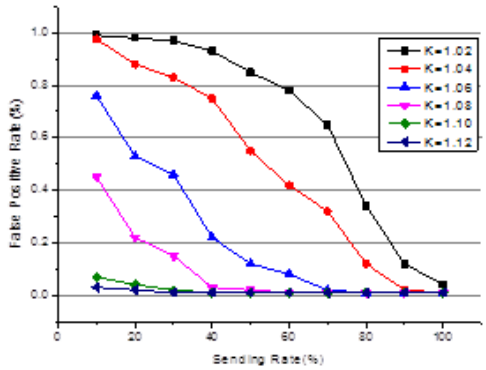
As you can see from Figure 5, smaller packet buffer requires smaller received power changes when compromised nodes are detected. A buffer with a length of 2 can reach a positive rate of 95% with the minimum change in received power. However, the false positive rate will be higher. For example, a buffer with a length of 2 has a false positive rate of 95%. These results can be explained by setting up a baseline with past data. A smaller buffer means that the algorithm is more sensitive to small changes, whether the change is caused by a compromise or a temporary change in the environment.

For the algorithm using the transmission rate, the buffer length L is 6 and the intercepted length L_2 is 2. Figure 6(a) and 6(b) show the performance of the algorithm when the received power is changed according to a certain percentage and threshold K. The result of this experiment is consistent with the results of previous experiments: smaller threshold and buffer length will make the algorithm more sensitive and provide higher detection rate at the expense of higher false positive rate. For example, if the value of K is 1.02, an increase of 30% of the transmission rate will make the detection rate up to 90%,

but the false positive rate is 97%. The detection time is only dependent on K and remains unchanged when the transmission rate changes.



(a) Detection rate



(b) False positive rate

Figure 6: Performance of detection algorithm based on sending rate change

The purpose of these experiments is to analyze the performance of the detection algorithms deployed on each node, and the actual parameters should be adjusted according to the needs of the application security. However, these results show that the algorithm can detect the compromised node with a detection rate of over 98% and a false positive rate below 5%.

In different environments, the performance of the algorithm will be quite different, because a single node with limited resources can not achieve high accuracy in any case. The function of the detection algorithm is to notify the base station of possible compromise behaviors. The base station determines whether a report is correct by collecting reports from multiple nodes, which will partly compensate for the limitations of the sensor nodes.

6.2 Overall Performance of CND

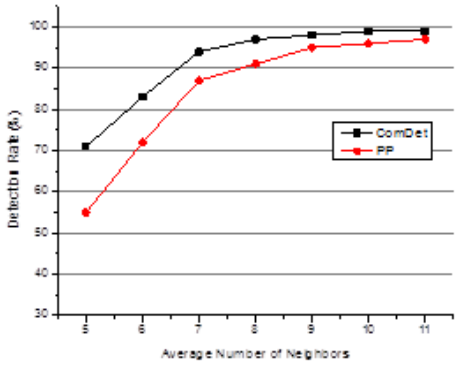
The ComDet system adopts a hybrid architecture consisting of two parts: distributed components and centralized components. Distributed components running on sensor nodes can detect compromised nodes and report them to the base station. Centralized components are used to perform complex analysis to determine whether the report of compromise is correct. The combination of these two aspects can effectively improve the efficiency and accuracy of detection. Document [6] proposes an intrusion detection scheme based on projection pursuit algorithm for wireless sensor networks, in which the proposed algorithm is called PP algorithm. Figure 7 shows a comparison between the ComDet algorithm and the PP algorithm in terms of detection rate, false positive rate and detection time when the packet loss rate is 15%. Through comparison, we can see that ComDet algorithm has higher detection efficiency and accuracy than BP algorithm.

In this paper, several experiments have been carried out to make a quantitative analysis of the performance of CND. The experimental setup is the same as that before, and 100 nodes that run TinyOS application are deployed randomly.

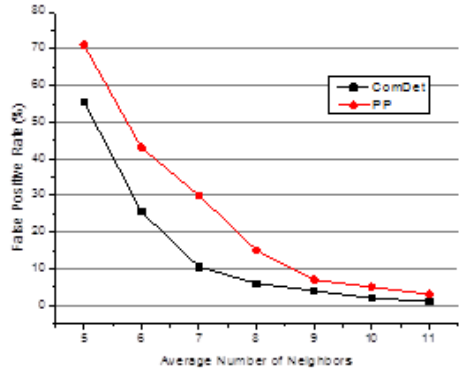
Figure 8 shows the performance evaluation results of CND with various packet loss rates and multiple monitoring neighbors. Figure 8(a) and 8(b) show that the algorithm can compensate for high packet loss rates when multiple nodes are monitored each other. When the packet loss rate is 30%, if each node monitors an average of 9 neighbors, it can reach a positive rate of 99% and a false positive rate of 2%. The high packet loss rate has a higher impact on the detection rate than the false positives, because the loss of the report makes the real compromise appear to be an instantaneous error. However, in most cases, the compromise can be detected before the damage is caused. As shown in figure 8(c), the higher the packet loss rate is, the longer the detection time is.

At the same time, the number of packets sent is also measured to be related to the operation of CND, as shown in Figure 8(d). As expected, the high packet loss rate will cause more packets to be sent because of retransmission. However, increasing the number of monitoring neighbors does not increase the number of packets sent. In most cases, the number of sending packets does not change significantly, and in some cases, as each neighbor is added, the number of actual packets sent by each neighbor is reduced by 5%. Careful observation shows that when the number of monitoring neighbors increases, more neighbors will send reports on the same attack, which will lead to increased communication overhead. However, more reports will make the base station detect compromised nodes faster when some reports are missing. The malicious behavior of the detected compromised node will no longer generate reports, which will reduce the communication overhead. The actual result is that, when the packet loss rate is greater than 15%, the communication overhead will decline or remain unchanged with each additional neighbor.

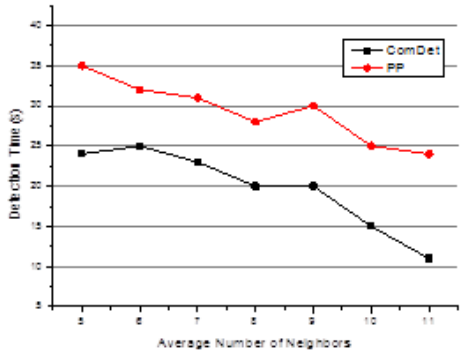
In addition, the energy consumption of CND is also measured. The energy consumption of wireless transmission accounts for the vast majority of the total energy consumption, which is consistent with the previous conclusion, that is, the total energy consumption is proportional to the communication overhead [11].



(a) Comparison of detection rate between ComDet and PP



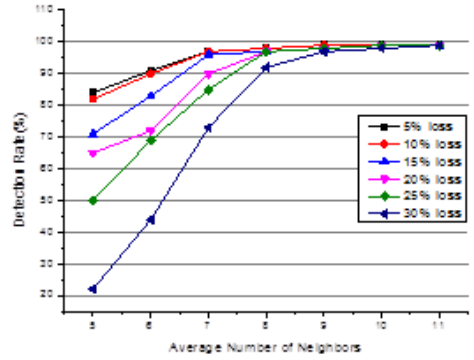
(b) Comparison of false positive rate between ComDet and PP



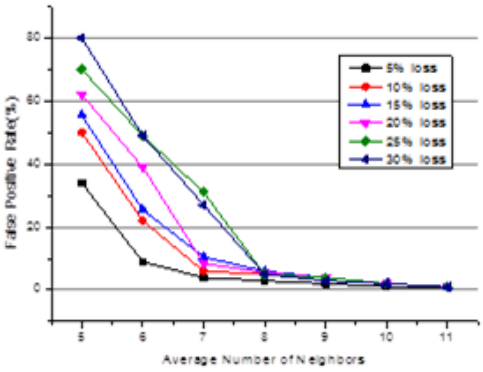
(c) Comparison of detection time between ComDet and PP

Figure 7: Performance comparison between ComDet and PP

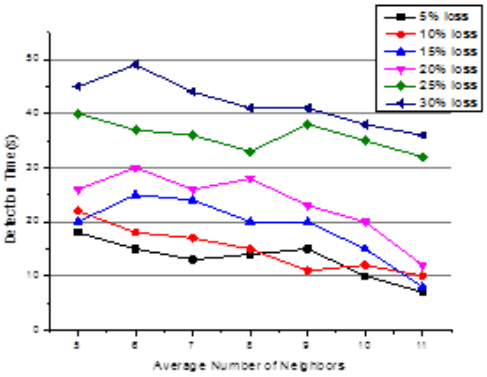
These results show that CND can provide accurate detection of compromised nodes and can be extended to large networks. Although similar systems can achieve almost the same performance without losing packets, but when the packet loss rate reaches 30%, the highest detection rate is reduced to 14%, and the false positive rate is as high as 99%. However, in the case of the packet loss rate of 30%, CND can reach a detection rate of 99% and a false positive rate of 2%. In addition, for a larger net-



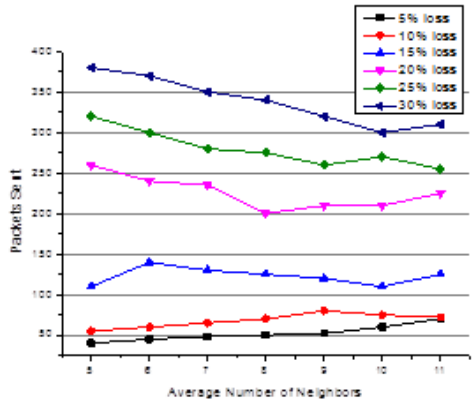
(a) Detection rate



(b) False positive rate



(c) Detection time



(d) Communication overhead

Figure 8: Performance of CND under various packet loss rates and neighbors

work of density and size, its overhead does not increase significantly.

7 Conclusion

In wireless sensor networks, compromised nodes can destroy data integrity by sending false reports, injecting erroneous data and interfering data transmission. Because encryption is not enough to prevent these attacks, CND is proposed to detect compromised nodes in wireless sensor networks. A series of experiments show that CND can reach a 99% detection rate and a false positive rate of less than 2% when the packet loss rate is 30%. CND can run in most wireless sensor networks, because it uses common application features and adjusts detection behavior when there is no periodic transmissions or lack of communications between nodes. It has smaller memory and lower computing and communication overhead, which enable it to be extended to large networks with thousands of nodes.

The goal of future work is to create a response system and a challenge system. CND provides a mean to identify compromised nodes in the network, but it does not provide a way to deal with attacks. The basic method is to isolate compromised nodes, but it is not suitable for all occasions. Besides, once a node is determined to be a compromised node, it should be allowed to prove that it is not a compromised node. This can further improve the accuracy of the detection.

Acknowledgments

This work was supported by the Experimental Technology Research Project Fund of Qufu Normal University (No. SJ201719), MOE (Ministry of Education in China) Project of Humanities and Social Sciences under Grant [No. 17YJCZH026].

References

[1] M. Amjad, M. Sharif, M. K. Afzal, S. W. Kim, "TinyOS - New trends, comparative views, and supported sensing applications: A review," *IEEE Sensors Journal*, vol. 16, no. 9, pp. 2865–2889, 2016.

[2] X. Chen, K. Makki, Y. Kang, and N. Pissinou, "Node compromise modeling and its applications in sensor networks," in *IEEE Symposium on Computers and Communications*, pp. 575–582, 2007.

[3] Y. Chen, and Q. Ye, "Summaryon security authentication scheme for wireless sensor networks," *Computer & Digital Engineering*, vol. 42, no. 2, pp. 261–266, 2014.

[4] A. R. Dhakne, P. N. Chatur, "Detailed survey on attacks in wireless sensor network," in *Proceedings of the International Conference on Data Engineering and Communication Technology*, Springer Singapore, 2017.

[5] Y. Gao, P. Zeng, K. K. R. Choo, and S. Fu, "An improved online/offline identity-based signature scheme for WSNs," *International Journal of Network Security*, vol. 18, no. 6, pp. 1143–1151, 2016.

[6] X. Ge, L.Wang, and X. Guo, "Intrusion detection model for WSNs based on projection pursuit," *Transducer & Microsystem Technologies*, vol. 34, no. 9, pp. 24–26, 2015.

[7] S. Javanmardi, A. Barati, S. J. Dastgheib, and I. Attarzadeh, "A novel approach for faulty node detection with the aid of fuzzy theory and majority voting in wireless sensor networks," *International Journal of Advanced Smart Sensor Network Systems*, vol. 2, no. 4, pp. 1–10, 2012.

[8] Z. Ji, X. Du, L. Xu, and J. Lin, "Security key predistribution scheme for wireless sensor networks," *Journal of Computer Applications*, vol. 33, no. 7, pp. 1851–1853, 2013.

[9] M. Kumar, K. Dutta, "Impact of wormhole attack on data aggregation in hierarchical WSN," *International Journal of Electronics & Information Engineering*, vol. 1, no. 2, pp. 70–77, 2014.

[10] C. T. Li, M. S. Hwang and Y. P. Chu, "An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, Aug. 2009.

[11] M. J. Miller and N. H. Vaidya, "A MAC protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228–242, 2005.

[12] A. Pananjady, V. K. Bagaria, R. Vaze, "Optimally approximating the coverage lifetime of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 98–111, 2017.

[13] Y. B. Saied, A. Olivereau, "A lightweight threat detection system for industrial wireless sensor networks," *International Journal of Network Security*, vol. 18, no. 5, pp. 842–854, 2016.

[14] R. Singh, M. S. Manu, "An energy efficient grid based static node deployment strategy for wireless sensor networks," *International Journal of Electronics & Information Engineering*, vol. 7, no. 1, pp. 32–40, 2017.

[15] L. Tan, and C. Pei, "Research of wireless sensors network security algorithms," *Computer Science*, vol. 42, no. S1, pp. 438–443, 2015.

[16] Y. Zhang, J. Yang, W. Li, and L. Jin, "An authentication scheme for locating compromised sensor nodes in WSNs," *Journal of Network & Computer Applications*, vol. 33, no. 1, pp. 50–62, 2010.

[17] J. Zhu, P. Wu, X. Wang, and J. Zhang, "SenSec: Mobile security through passive sensing," in *IEEE International Conference on Computing, Networking and Communications*, pp. 1128–1133, 2013.

Biography

Xiaolong Xu, born in 1977, he is now a experimenter of Experiment Teaching Center, Qufu Normal University. He obtained his bachelor's degree of computer science from Qufu Normal University in 1999, and master's degree of computer application technology from Qufu Normal University in 2006. He has published more than 20 papers. His major research interests include network security and wireless sensor network.

Zhonghe Gao, born in 1961. He is a professor and postgraduate tutor. His major research interests include computer network and mobile communication technology.

Lijuan Han, born in 1976. She got the master degree and now is a associate professor. Her major research interests include artificial intelligence and digital signal processing.