sensors

MDPI

Article

A Protocol Layer Trust-Based Intrusion Detection Scheme for Wireless Sensor Networks

Jian Wang [1,*], Shuai Jiang [1] and Abraham O. Fapojuwo [2]

[1] School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China; jiangshuai93@163.com

[2] Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; fapojuwo@ucalgary.ca

* Correspondence: jwang@nudt.edu.cn; Tel.: +86-731-8457-4481

Academic Editors: Jaime Lloret Mauri and Guangjie Han
Received: 29 March 2017; Accepted: 24 May 2017; Published: 27 May 2017

Abstract: This article proposes a protocol layer trust-based intrusion detection scheme for wireless sensor networks. Unlike existing work, the trust value of a sensor node is evaluated according to the deviations of key parameters at each protocol layer considering the attacks initiated at different protocol layers will inevitably have impacts on the parameters of the corresponding protocol layers. For simplicity, the paper mainly considers three aspects of trustworthiness, namely physical layer trust, media access control layer trust and network layer trust. The per-layer trust metrics are then combined to determine the overall trust metric of a sensor node. The performance of the proposed intrusion detection mechanism is then analyzed using the t-distribution to derive analytical results of false positive and false negative probabilities. Numerical analytical results, validated by simulation results, are presented in different attack scenarios. It is shown that the proposed protocol layer trust-based intrusion detection scheme outperforms a state-of-the-art scheme in terms of detection probability and false probability, demonstrating its usefulness for detecting cross-layer attacks.

Keywords: protocol layer; intrusion detection; wireless sensor networks

1. Introduction

Wireless sensor networks (WSNs) are widely used in variety of fields, such as agricultural, environmental, industrial and military monitoring applications. WSNs can be divided into flat- architecture WSNs and clustered-architecture WSNs according to their architecture. In the flat- architecture WSNs, all the sensor nodes (SNs) transmit their own data or relay data for other nodes to the base station (BS). In the clustered-architecture WSNs, the adjacent SNs are organized as a cluster, and a cluster head (CH) controls its cluster. The SNs that belong to the same cluster can only exchange information via their CH and the CH transmits information either directly or through some other CHs to the BS. Clustering brings many advantages, such as energy efficiency, better network communication, efficient topology management, minimized delay, and so on.

WSNs face serious security problems, because of the openness of nodal deployment and wireless communication. In some WSN deployments, the SNs may be captured and the key information might be leaked or compromised. The purpose of an attacker is to disrupt the security attributes of WSNs, including confidentiality, integrity, availability and authentication. To achieve these objectives, the attacker may launch attacks from different protocol layers of WSNs. At the physical layer, the attacker can jam the physical channel by interfering with the radio frequencies that nodes use for communication [1]. The attacker can also extract the secret information from the captured node, tamper with its circuitry, modify the program codes, or even replace it with a malicious node [2]. Attacks at the medium access control (MAC) layer aim to disrupt the availability of the network by

purposefully creating collisions, obtain unfair priority in the contention for the channel or dissipate the limited energy of nodes. Attacks at the MAC layer include collision, denial of sleep, Guaranteed Time Slot (GTS) attack, back-off manipulation and so on [3–6]. Attacks at the network layer aim to disrupt the network routing, and acquire or control the data flows. Examples are spoofed routing information, selective packet forwarding, sinkhole, wormhole, blackhole, sybil, and hello flood attack [7–10]. Besides the attacks aiming at a single protocol layer, there are cross-layer attacks which relate to multiple layers in WSNs [11–13]. Cross-layer attack can achieve better attack effects, better conceal the attack behavior or reduce the cost of attack compared to the attacks at a single layer.

Considering the limited resources of the SNs, it is not realistic for WSNs to implement high-strength security mechanisms. Furthermore, the attacker may have the ability of breaking through or bypassing the protection of security mechanism with the progress of attack technologies. Thus serving as a second wall, intrusion detection plays an important role in protecting the network. The intrusion detection system for WSNs can detect whether there are behaviors violating the security policy and record evidence of being attacked by collecting and analyzing the information from sensor nodes and networks. It can send alarm timely to the system administrator and perform some countermeasures against the attack.

There are now two kinds of intrusion detection systems [14]. One is the misuse detection system, the other is the anomaly detection system. Misuse detection is based on predefined rules, where it is easy to detect known attacks, but impossible to detect unknown attacks. Anomaly detection compares present activities with normal system status and user behaviors to detect anomalies. Compared with misuse detection, anomaly detection has higher detection rate and the ability to detect unknown attacks, with its false positive rate increasing correspondingly. The focus of this paper is on anomaly detection schemes. Recently, different types of anomaly detection schemes based on traffic prediction [15], statistical method [16], data mining [17], game theory [18–20], immune theory [21], or trust management [22–37], etc., have been proposed.

However, there are still some unsolved issues in the existing intrusion detection schemes for WSNs. Many of the schemes detect attacks according to the anomalies of network traffic. Actually, it is a great challenge to distinguish normal behavior from abnormal behavior because not all of attacks on WSNs will introduce abnormal network traffic. Many intrusion detection schemes only aim to detect several typical types of attacks, while the scenarios of different types of attacks carried out concurrently or cross-layer attacks are seldom considered. The attack behaviors on WSNs are usually interconnected and transformed mutually. It is difficult to obtain good detection performance by only studying how to detect a certain kind of attack. Therefore, it is necessary to pay more attention to complex attack behaviors, such as cross-layer attack, and study how to utilize the protocol feature parameters at different protocol layers, especially the key parameters which may have an important influence on the performance of the network in order to improve the detection ability of intrusion detection systems [38].

In this paper, we propose a trust-based intrusion detection scheme which uses the deviations of parameters of multiple protocol layers as trust metrics, considering that the attacks will inevitably have impacts on the parameters of the different protocol layers. Inspired by the method proposed by Bao et al. [34,35], we utilize weighting method to build the system model and t-distribution to analyze the performance of our scheme. In our scheme, the monitoring node observes the key parameters of the monitored nodes at the physical layer, MAC layer and network layer, and calculates the deviations of these key parameters. According to the deviations of the parameters, the monitoring node can evaluate the trustworthiness toward the monitored node by aggregating the trust values at different layers and send it to the CH or BS. The CH or BS can then calculate the aggregated trust value of a node according to the trust values which are evaluated by multiple monitoring nodes. If the trust value of a node is less than a predefined threshold, the node is regarded as abnormal. Because the key parameters of multiple layers are being monitored, it is effective for our scheme to detect different

types of attacks at different protocol layers. Moreover, our scheme is applicable to both clustered WSNs and flat WSNs.

The rest of this paper is organized as follows. Section 2 surveys existing work on trust-based intrusion detection in WSNs. Section 3 describes our intrusion detection scheme. Section 4 analyzes the performance of our scheme by using analytical and simulation approaches, and compares its performance results with those of an existing scheme in the literature. Section 5 concludes the paper.

2. Related Work

Trust management is an effective method to identify malicious, selfish or compromised nodes. In recent years, research on trust management and its application to intrusion detection has received considerable attention from researchers. The current trust evaluation schemes aim to improve the detection performance, resource efficiency, robustness etc., by using fuzzy theory, probability theory and statistics, weighting method, etc. [22].

In [23–25], fuzzy theory is used to determine the trust degree of a sensor node. Feng et al. [23] proposed a trust evaluation algorithm named as Node Behavioral strategies Banding belief theory of the Trust Evaluation algorithm (NBBTE). In their scheme, each node firstly establishes the direct and indirect trust values of neighboring nodes by comprehensively considering various trust factors and then fuzzy set theory is used to decide the trustworthiness levels of the sensor nodes. Finally, D–S evidence theory method is adopted to obtain an integrated trust value instead of a simple weighted-average one. Wu et al. [24] put forward a trust model to detect anomaly nodes in WSNs based on fuzzy theory and evidence theory. Fuzzy theory is used to calculate the trustworthiness levels of multi-dimensional characteristics of the evaluated node and the evidence theory is applied to integrate a direct trust value for the evaluated node. Shao et al. [25] proposed a lightweight and dependable trust model for clustered wireless sensor network, in which the fuzzy degree of nearness is adopted to evaluate the reliability of the recommended trust values from the third party nodes.

In [26,27], probability distribution is used to build the trust evaluation model. Ganeriwal et al. [26] presented a distributed reputation-based framework for sensor networks. It uses a watchdog mechanism to monitor communication behaviors of neighboring nodes, represents node reputation distribution using Beta distribution and calculates the trust value according to the statistical expectation of the probability reputation distribution. Luo et al. [27] proposed a dynamic trust management scheme for WSNs. It uses a hash algorithm to generate identify labels for SNs and builds a trust-evaluating model based on beta density function.

In [28–33], trust is estimated using weighting method. Atakli et al. [28] proposed a weighted-trust evaluation based scheme to detect compromised or misbehaved nodes in WSNs by monitoring their reported data. The hierarchical network can reduce the communication overhead between sensor nodes by utilizing clustered topology. Shaikh et al. [29] presented a group-based trust management scheme for clustered WSNs. It evaluates the trust of a group of nodes in contrast to traditional trust schemes that usually focus on the trust values of individual nodes, which reduces the cost of trust evaluation. Yao et al. [30] put forward a parameterized and localized trust management scheme for WSNs, where each sensor node maintains highly abstracted parameters, rates the trustworthiness of its interested neighbors to adopt appropriate cryptographic methods, identify the malicious nodes, and share the opinion locally. Li et al. [31] proposed a lightweight and dependable trust system for clustered WSNs. Given the cancellation of feedback between nodes, it can greatly improve system efficiency while reducing the effect of malicious nodes. By adopting a dependability-enhanced trust evaluating approach for cooperation between CHs, it can effectively detect and prevent malicious, selfish and faulty CHs. Jiang et al. [32] presented an efficient distributed trust model for WSNs. In their model, the trustworthiness of a node includes direct trust and indirect trust. During the calculation of direct trust, communication trust, energy trust and data trust are considered. When a subject node cannot directly observe object nodal communication behaviors, the indirect trust value is gained based on the recommendations from some other nodes. Ishmanov et al. [33] put forward a lightweight

and robust trust establishment scheme using the weight of misbehavior. In their scheme, a new trust component, misbehavior frequency is introduced to improve the resiliency of the trust mechanism.

Bao et al. [34,35] utilizes weighting method to build the trust evaluation model and statistical method to analyze the false alarm probability. In [34], they presented a trust-based intrusion detection scheme using a highly scalable cluster-based hierarchical trust management protocol. It considers both quality of service trust and social trust as trust metrics and uses an analytical model based on stochastic Petri nets to evaluate the performance of the scheme, as well as a statistical method to calculate the false alarm probability. They adopt honesty to measure social trust and energy and cooperativeness to measure quality of service trust. In [35], intimacy, honesty, energy, and unselfishness are considered as four different trust components.

In [36,37], some new models are used to evaluate the trustworthiness. Zhang et al. [36] put forward a trust evaluation method for clustered wireless sensor networks based on cloud model, which implements the conversion between qualitative and quantitative of trust metrics and produces different types of trust cloud to evaluate trust values of cluster heads and cluster members. Rajeshkumar et al. [37] presented a trust based adaptive acknowledgment intrusion detection system for WSNs based on number of active successful deliveries, and Kalman filter to predict node trust.

It is important for a trust management scheme to select proper trust factors to evaluate the trustworthiness of a SN. From the literature on this topic, we can find that the trust factors of a SN is mainly based on the nodal communication behavior, energy level, or recommendation from the third party and there is no unified standard in the selection of the trust factors. The attacks initiated at each protocol layer and their influence on the parameters of the corresponding protocol layers lack comprehensive analysis. To the best of our knowledge, there is still no trust management scheme which elaborately describes the trustworthiness of a SN from the standpoint of protocol layer. Thus, it is interesting to build the trust evaluation model based on the protocol layer trust. In view of the reality of intrusion detection scheme, we mainly consider the direct trust of a node in our scheme and the trustworthiness of a node is evaluated according to its behaviors at different protocol layers. The consideration of trust worthiness from the viewpoint of multiple protocol layers distinguishes this paper from the previous related works [23–37]. Since the deviations of the key parameters of multiple layers are used to evaluate the trustworthiness of a node, it is helpful for our scheme to detect nodal malicious behaviors initiated from different protocol layers, which is effective for detecting cross-layer attacks.

3. System Model

We consider a WSN where the network can be divided into multiple clusters, as illustrated in Figure 1. Each cluster consists of a number of SNs and a CH. SNs can communicate with their CH either directly or through other SNs. A CH can forward the aggregated data to the BS directly or through other CHs.

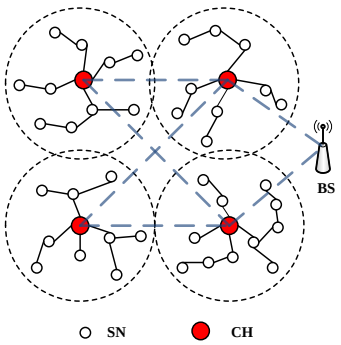

Figure 1. Clustered wireless sensor networks (WSN) architecture. (SN = sensor node, CH = cluster head, BS = base station).

Our trust-based intrusion detection scheme includes two levels of trust evaluation, one is CH-to-SN trust evaluation, the other is BS-to-CH trust evaluation. In CH-to-SN trust evaluation, each SN evaluates its neighbors and sends the trust evaluation results to its CH periodically. The CH evaluates all the SNs in its cluster by analyzing statistically the trust evaluation results reported by other SNs. The trust update period is Δt, which is a system parameter. The length of Δt could be made shorter or longer based on network analysis scenarios. Similarly, in BS-to-CH trust evaluation, each CH performs trust evaluation toward its neighboring CHs and sends its trust evaluation results to the BS. The BS evaluates all the CHs in the network by using the same methods as adopted in CH-to-SN trust evaluation. Since the two levels of trust evaluation use the same method, we mainly describe CH-to-SN trust evaluation.

The nodal trustworthiness consists of the trust degree of each protocol layer, including physical layer, MAC layer, network layer, transport layer and application layer. Since most of the attacks against WSNs aim at the physical layer, MAC layer and network layer, for simplicity, in this paper we mainly focus on the trusts at these three layers. Let $T_{ij}^{DIRECT}(t)$ denote the trust value that the sensor node i directly evaluates toward its neighboring node j at time t. It can be calculated by:

$$T_{ij}^{DIRECT}(t) = w_1 T_{ij}^{PHY}(t) + w_2 T_{ij}^{MAC}(t) + w_3 T_{ij}^{NET}(t), \quad (1)$$

where $T_{ij}^{PHY}(t)$, $T_{ij}^{MAC}(t)$, and $T_{ij}^{NET}(t)$ represents the trust value that node i evaluates toward node j at the physical (PHY) layer, medium access control (MAC) layer, and network (NET) layer, respectively, w_1, w_2, and w_3 are the corresponding weight values associated with these three trust components, $w_1 \in [0,1]$, $w_2 \in [0,1]$, $w_3 \in [0,1]$ and $w_1 + w_2 + w_3 = 1$. The values of the weights w_1, w_2, and w_3 are determined according to the concrete requirement of a detection system under implementation. Generally speaking, the number of attacks aiming at the network layer is greater than those aiming at the MAC layer and physical layer. Hence, the value of w_3 is usually slightly larger than that of w_1 or w_2. In order to evaluate the trustworthiness of each protocol layer, we can choose some important parameters at each protocol layer and calculate the deviations of these parameters. Actually, our scheme is scalable. If a more accurate trust value is needed, we can choose additional parameters at each protocol layer and calculate the deviations of these parameters. Certainly, the more parameters are selected, the more complex the detection system will be. Hence, we can select parameters according to the requirement and complexity of the detection system.

The trustworthiness of a SN (or CH) should be updated periodically. Node i evaluates the trust of node j during a time window of length Δt, so the updated trust of node i toward node j is:

$$T_{ij}(t) = \alpha T_{ij}(t - \Delta t) + (1 - \alpha) T_{ij}^{DIRECT}(t), \quad (2)$$

where $T_{ij}(t - \Delta t)$ denotes the historical trust value of node i toward node j, and $\alpha \in [0,1]$ is the weight value of the historical trust value. Actually, the direct observation result is more important and accurate than the historical trust value. Therefore, α can be defined as $e^{-\Delta t}$. Next, we will describe the calculation of the trust at each protocol layer.

3.1. Calculation of Physical Layer Trust

Energy consumption rate is an important parameter at the physical layer. A malicious node usually sends or receives more packets than a normal node. It will inevitably consume more node energy, so we choose energy consumption as trust metric at this layer. The monitoring node i can obtain the energy consumption of its neighboring node j during the time period of Δt. The relative deviation of energy consumption of node j can be calculated by:

$$RD_{EC}(t) = \frac{\Delta E_j(t) - \overline{\Delta E(t)}}{\overline{\Delta E(t)}}, \quad (3)$$

in which $\Delta E_j(t) = E_j(t - \Delta t) - E_j(t)$, $\overline{\Delta E(t)} = \frac{1}{n}\sum_{i=1}^{n} \Delta E_i(t)$. $E_j(t)$ indicates the residual energy of node j at time t and $\Delta E_j(t)$ represents the energy consumption of node j during the time period of Δt. $\overline{\Delta E(t)}$ is the average energy consumption level of all neighboring nodes of node i during this time period and n denotes the number of neighboring nodes of node i. Node i can roughly evaluate the energy consumption of its neighboring nodes during the time period of Δt by monitoring their packet transmission activities. The greater the deviation of energy consumption is, the lower the nodal trustworthiness will be. So we obtain the physical layer trust as:

$$T_{ij}^{PHY}(t) = \begin{cases} 1 - RD_{EC}(t), \text{ if } 0 < RD_{EC}(t) < 1 \\ 0, \ RD_{EC}(t) \geq 1 \\ 1, \ RD_{EC}(t) \leq 0 \end{cases}, \quad (4)$$

In Equation (4), if the relative deviation of energy consumption is less than or equal to 0, which means the energy consumption of the monitored node is less than the average energy consumption, the monitored node is considered trustworthy at the physical layer. If RD_{EC} is greater than or equal to 1, which means the energy consumption of the monitored node is more than double or double the average energy consumption, the monitored node will be considered untrustworthy at the physical layer.

3.2. Calculation of MAC Layer Trust

Next we calculate the MAC layer trust. There are variety of attacks initiated at this layer whose main objective is to get the priority of channel access. A malicious node can select a small back-off time, choose a small size of contention window (CW), or wait for shorter interval than distributed inter-frame spacing (DIFS), aiming to gain significant advantage in the contention of channel over the unmalicious nodes. Therefore, the interval time between two consecutive successful transmissions of malicious node, which we define as idle time, will be less than that of the unmalicious node. The malicious node can also scramble the frames sent by other nodes in order to obtain the priority of channel access. As a result, the average number of retransmissions of the malicious node will be less than that of the unmalicious node. As described above, we choose two important parameters, the idle time and number of retransmissions, as the trust metrics at the MAC layer. Thus, at the MAC layer, the node i evaluates the trust value of node j as:

$$T_{ij}^{MAC}(t) = p_1 T_{ij}^{idle\_time}(t) + p_2 T_{ij}^{num\_retr}(t), \quad (5)$$

where p_1, p_2 are the weight values associated with the two trust components, $p_1 \in [0,1]$, $p_2 \in [0,1]$, and $p_1 + p_2 = 1$. The exact values of p_1 and p_2 depend on the requirements of the detection system under implementation.

In order to calculate $T_{ij}^{idle\_time}(t)$, the monitoring node i can obtain the idle time x_k (k means the k-th transmission of the monitored node) according to Request To Send (RTS)/Clear To Send (CTS) access in Distributed Coordination Function (DCF) mode, and x_k can be calculated by:

$$x_k = t_k - t_{k-1} - t_{SIFS} - t_{ACK}, \quad (6)$$

where t_k denotes the time of the k-th RTS packet reception, t_{k-1} is the end time point of the reception of the previous data segment, t_{SIFS} is the duration of the Short Inter-Frame Spacing (SIFS) frame, and t_{ACK} is the duration of Acknowledgement (ACK) frame, as illustrated in Figure 2.

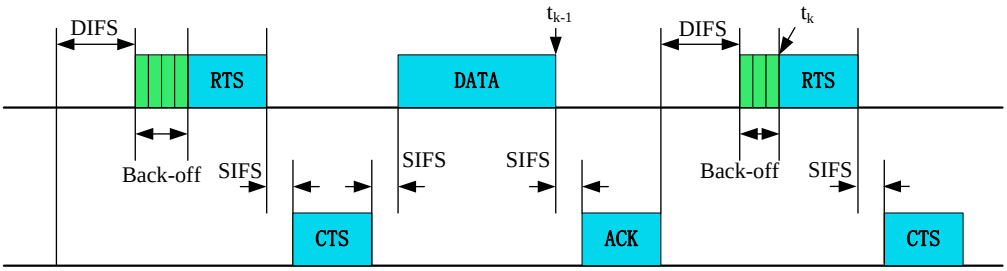

Figure 2. 802.11 Distributed Coordination Function (DCF) operation. RTS = Request to Send, ACK = Acknowledgement, CTS = Clear To Send, DIFS= distributed inter-frame spacing, SIFS = short inter-frame spacing.

For an unmalicious node, $x_k = t_{DIFS} + b_k$, where t_{DIFS} is the duration of DIFS frame, and b_k is the random back-off time. A malicious node is trying to decrease the idle time by manipulating the back-off time and DIFS period. Therefore, the monitoring node can detect these misbehaviors by calculating the deviation of the idle time. We can obtain the average idle time of the CH, according to:

$$\bar{x} = t_{DIFS} + \bar{b} = t_{DIFS} + \frac{1}{u}\sum_{k=1}^{u} b_k, \tag{7}$$

where u denotes the number of successful transmissions by the CH during the observation period of Δt. We then calculate the deviation of the idle time:

$$D_{idle\_time}(t) = \frac{1}{m}\sum_{k=1}^{m}(x_k - \bar{x}), \tag{8}$$

where m is the observed number of successful transmissions of the monitored node. Therefore, the relative deviation of the idle time can be expressed as:

$$RD_{idle\_time}(t) = \frac{|D_{idle\_time}(t)|}{\bar{x}} \tag{9}$$

and the idle time trust is calculated by:

$$T_{ij}^{idle\_time}(t) = \begin{cases} 1 - RD_{idle\_time}(t), \text{ if } \sum_{k=1}^{m}(x_k - \bar{x}) < 0 \\ 1, \text{ else} \end{cases}. \tag{10}$$

It means that the trust value of the monitored node will decrease if its idle time is less than the average idle time.

In order to calculate the number of retransmissions trust $T_{ij}^{num\_retr}(t)$, we first calculate the deviation of the number of retransmissions of the monitored node j. The monitoring node i can detect a retransmission by observing a repeated sequence number in the head of frames. It monitors the number of retransmissions of node j during the time period of Δt, which is denoted by $y_{ij}(t)$. It can also obtain the average number of retransmissions $\overline{y(t)}$ during the time period of Δt, by monitoring the number of retransmissions of its neighboring nodes. $\overline{y(t)} = \frac{1}{n}\sum_{k=1}^{n} y_{ik}(t)$, where $y_{ik}(t)$ means the number of retransmissions of node k during the time period of Δt, node k is one of the neighboring nodes of node i and n denotes the number of neighboring nodes of node i. Then, the relative deviation of the number of retransmissions of node j can be calculated by:

$$RD_{num_{retr}}(t) = \frac{\overline{y(t)} - y_{ij}(t)}{\overline{y(t)}} \tag{11}$$

and the number of retransmissions trust can be expressed as:

$$T_{ij}^{num\_retr}(t) = \begin{cases} 1 - RD_{num_{retr}}(t), \text{ if } y_{ij}(t) < \overline{y(t)} \\ 1, \text{ else} \end{cases}. \quad (12)$$

If the number of retransmissions of node j is less than the average number of retransmissions, its trust value will decrease.

3.3. Calculation of Network Layer Trust

Attacks at the network layer aim to disrupt the network routing, and acquire or control the data flows. A malicious node can make itself a part of a routing path by advertising bogus routing messages, such as a good Link Quality Indicator (LQI) or a small hop count. It can also initiate sinkhole or selective forwarding attack and result in dropping all or part of forwarding packets. Therefore, we choose route metric and packet forwarding rate as trust metrics to evaluate the network layer trust. The network layer trust is described as:

$$T_{ij}^{NET}(t) = q_1 T_{ij}^{route\_metric}(t) + q_2 T_{ij}^{PFR}(t), \quad (13)$$

where $q_1 \in [0,1]$, $q_2 \in [0,1]$ are weight values and $q_1 + q_2 = 1$. The exact values of q_1 and q_2 depend on the requirements of the detection system under implementation.

There are different route metrics for routing protocols in WSNs. For example, in the MintRoute protocol, it uses link estimates as routing metric and includes the LQI within its route update packet [39]. In the TinyAODV (Tiny Ad-hoc On-Demand Vector) protocol, the routing metric is the number of hop count and includes the hop count in Route Reply (RREP) packet [40]. A malicious node can make its neighbors change their current parents and choose it as their new one by advertising an attractive LQI for itself in the route update packet or giving a small value of hop count in RREP packet. We then take LQI and hop count as basis to calculate the route metric trust.

We can calculate the deviation of LQI by comparing the actual LQI value with the advertised one. When a monitoring node receives a route update packet from a monitored node, it can calculate the actual LQI value according to $LQI'_k = 255 \times (RSSI_k + 81)/91$ [41], where k denotes the k-th route update packet that it received and $RSSI_k$ represents the received signal strength indicator of the k-th route update packet. The monitoring node can obtain the advertised LQI from the route update packet which is denoted by LQI_k. Then the average deviation of LQI is calculated by:

$$D_{LQI}(t) = \frac{1}{m}\sum_{k=1}^{m}\left(LQI_k(t - \Delta t,\ t) - LQI'_k(t - \Delta t,\ t)\right), \quad (14)$$

where m denotes the number of route update packets that the monitoring node has received during the time period of Δt. Therefore, the LQI trust that node i evaluates toward node j can be described as:

$$T_{ij}^{LQI}(t) = \begin{cases} 1 - \frac{D_{LQI}(t)}{LQI_{max}}, \text{ if } \sum_{k=1}^{m}\left(LQI_k - LQI'_k\right) > 0 \\ 1, \text{ else} \end{cases}, \quad (15)$$

where LQI_{max} equals to 255 in MintRoute protocol [39]. This formula means that the trust degree of the monitored node will decrease if the advertised LQI value is larger than the actual one.

If the route metric is hop count, the monitoring node can also evaluate the trust degree of the monitored node by calculating the deviation of hop count. The monitoring node can calculate the average hop count toward destination node according to the RREP packets it has received during the time period of Δt. The average hop count is described as $\overline{hop\_count} = \frac{1}{n}\sum_{k=1}^{n} hop\_count_k$, where n denotes the number of received RREP packets during the observation time and hop_count_k is the value of hop count to the destination node, which is included in the k-th RREP packet. We can also

adopt the method in [42]. Each node builds a node neighbor database which contains the ID of the neighboring node and the hop count to the CH for each node. Thus, we can also calculate the average hop count by $\overline{\text{hop\_count}} = \frac{1}{n}\sum_{k=1}^{n} \text{hop\_count}_k$, where n denotes the number of neighboring nodes. The relative deviation of hop count of the monitored node j can be calculated by:

$$\text{RD}_{\text{hop}_{\text{count}}}(t) = \frac{\overline{\text{hop\_count}} - \text{hop\_count}_j}{\overline{\text{hop\_count}}}, \tag{16}$$

where hop_count_j denotes the value of hop count from node j to its CH. The hop count trust is described as:

$$T_{ij}^{\text{hop\_count}}(t) = \begin{cases} 1 - \text{RD}_{\text{hop}_{\text{count}}}(t), & \text{if } \text{hop}_{\text{count}j} < \overline{\text{hop\_count}} \\ 1, & \text{else} \end{cases} \tag{17}$$

This means if hop_count_j is less than the average hop count, the more deviation there is, and the lower the trust value will be.

In order to obtain the packet forward trust, the monitoring node i can obtain the packet forwarding rate of the monitored node j by:

$$T_{ij}^{\text{PFR}}(t) = \frac{P_{j\to k}(t)}{P_{i\to j\to k}(t)}, \tag{18}$$

where $P_{i\to j\to k}(t)$ denotes the number of packets that node i wants to transmit to node k with the help of node j and $P_{j\to k}(t)$ indicates the number of packets that node j has received from node i and forwarded to node k. If node j does not forward packets correctly, its trust degree will decrease.

In order to decide whether or not a node is considered compromised, it is necessary to select a system trust threshold, Th_{trust}. In a cluster, all of the monitoring nodes will send their trust evaluation results with respect to their neighboring nodes to their CH. The CH then computes the trust value of node j according to:

$$T_{cj}(t) = \frac{1}{n}\sum_{i=1}^{n} T_{ij}(t), \tag{19}$$

where n denotes the number of neighboring nodes of node j and makes decision by comparing the trust value with Th_{trust}. If $T_{cj}(t)$ is less than Th_{trust}, then node j is regarded as compromised. The method of BS-to-CH trust evaluation is similar to that of CH-to-SN trust evaluation.

4. Performance Analysis

The purpose of the analysis is to derive mathematical results of the false positive and false negative probabilities. The false positive probability is the probability that a node is evaluated as compromised whereas it is not. On the other hand, false negative probability is the probability that a node is evaluated as not compromised whereas it is. The expressions for the false positive and false negative probabilities are derived using a statistical approach. We also calculate the communication overhead of our scheme.

4.1. Statistical Analysis

We utilize t-distribution to analyze the performance of our trust-based intrusion detection scheme because it is suitable to detect the difference between two means in the circumstance of limited samples, which is similar to [34]. $T_{cj}(t)$ is a random variable with normal distribution and the standard deviation of $T_{cj}(t)$ is unknown. In order to calculate the false positive and false negative probabilities, we then transform $T_{cj}(t)$ into a random variable $X_j(t)$ following t-distribution with n-1 degrees of freedom, which is denoted by:

$$X_j(t) = \frac{T_{cj}(t) - \mu_j(t)}{S_j(t)/\sqrt{n}}, \tag{20}$$

where $T_{cj}(t) = \frac{1}{n}\sum_{i=1}^{n} T_{ij}(t)$ is the sample mean, $\mu_j(t)$ is the population mean of the trust value of node j, $S_j(t) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(T_{ij}(t) - T_{cj}(t)\right)^2}$ is the standard deviation of the trust value that node i evaluated with respect to node j, and n is the number of neighboring nodes of node j. We can obtain $\mu_j(t)$ by running simulations for many times. Thus, according to Equation (20), the probability that node j is evaluated as a compromised node is given by:

$$P(\mu_j(t) < Th_{trust}) = P\left(X_j(t) > \frac{T_{cj}(t) - Th_{trust}}{S_j(t)/\sqrt{n}}\right). \tag{21}$$

The false positive probability can be calculated by:

$$\rho^{fp}(t) = P(X_j(t) > \frac{T_{cj}^{N}(t) - Th_{trust}}{S_j^{N}(t)/\sqrt{n}} = \gamma) = \int_{\gamma}^{\infty} \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})}(1 + \frac{x^2}{n})^{-\frac{n+1}{2}}\,dx, \tag{22}$$

where $T_{cj}^{N}(t)$ ($S_j^{N}(t)$) is the mean value (standard deviation) under the condition that node j is not compromised, superscript N denotes Not compromised and $\Gamma(x) = \int_0^{\infty} t^{x-1}e^{-t}\,dt$ is the gamma function.

The false negative probability is expressed as:

$$\rho^{fn}(t) = P(X_j(t) \leq \frac{T_{cj}^{C}(t) - Th_{trust}}{S_j^{C}(t)/\sqrt{n}} = \delta) = \int_{-\infty}^{\delta} \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})}(1 + \frac{x^2}{n})^{-\frac{n+1}{2}}\,dx \tag{23}$$

where $T_{cj}^{C}(t)$ ($S_j^{C}(t)$) is the mean value (standard deviation) under the condition that node j is compromised, superscript C denotes Compromised.

4.2. Numerical Results and Discussion

We use Matlab as simulation tool to generate the performance results of our scheme. We consider a WSN with 50 nodes, randomly deployed in a 100 m × 100 m operational area. The transmitting power of a SN is 2 mW and the communication frequency is 2.4 GHz. The trust update interval is set to 10–100 min. The detailed simulation parameters are listed in Table 1.

Table 1. Assumed values of system parameters. MAC = medium access control, AODV = Ad-hoc On-Demand Vector.

Parameter	Value	Parameter	Value
Size of network	100 m × 100 m	w_1, w_2, w_3	1/3
Number of SNs	49	Number of CH	1
Trust update interval	10–100 min	p_1, p_2, q_1, q_2	1/2
MAC protocol	802.11 DCF	Routing protocol	AODV
t_{DIFS}	50 µs	t_{SIFS}	10 µs
Slot time	20 µs	t_{ACK}	112 µs
Size of Route Request (RREQ) packet	176 bits	Size of Route Reply(RREP) packet	176 bits

Figure 3 shows the relationship between the trust value of a SN and the simulation time and compares the trust value of the SN with the node density varying from 30 nodes to 50 nodes per 10,000 m^2 (e.g., 30 nodes mean 1 CH and 29 SNs). We observe that the trust value of the SN fluctuates in a narrow range (0.982, 0.984), when the simulation time is relatively short. If the simulation time is long enough, the trust value of the SN becomes stable, because the longer the simulation time is, the more data are collected and the more accurate the results are. We also notice that with the increase of node density the fluctuation of the trust value of the monitored node becomes smaller. This is because if the node density is small, the number of neighboring nodes of the monitored node will

be small, the data that the monitoring node can obtain will be less and hence the trust value of the monitored node will not be so accurate.

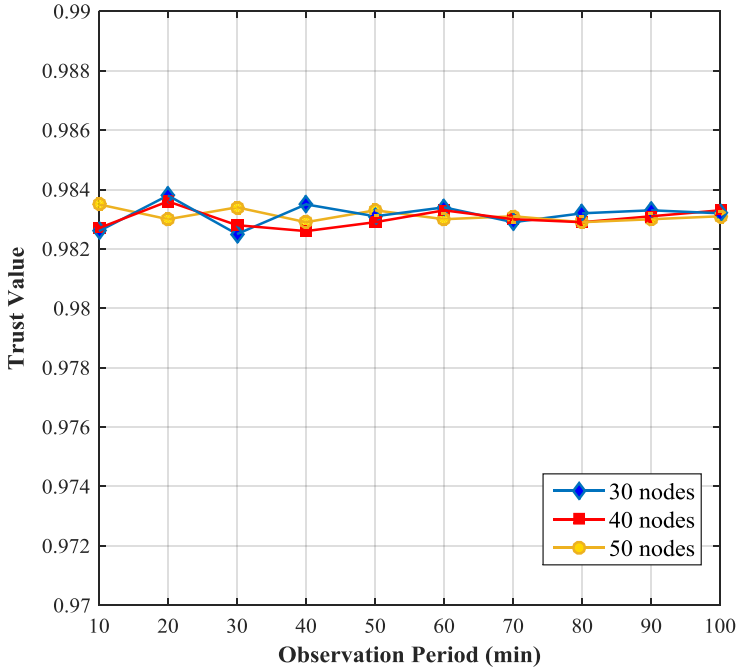

Figure 3. Sensitivity of trust value to simulation time and node density.

Figure 4 shows the variation of the trust values of the monitored node under the scenario of several types of attacks. We simulate four typical attacks at the MAC layer and network layer, including back-off manipulation, selective forwarding attack, sinkhole attack and MAC-Network cross-layer attack. In the back-off manipulation attack, a malicious node gets unfair priority access to the channel by setting a small CW. In the selective forwarding attack, a malicious node selectively drops packets passing though it according to a predefined criterion. In the sinkhole attack, an attacker tries to attract network traffic by sending bogus RREP messages. In the MAC-Network cross-layer attack, the malicious node initiates attacks at the MAC layer and network layer simultaneously to make itself a node on the routing path by using a small CW and sending a fake routing message with small hop count. We observe that if a node initiates attacks its trust value will decrease obviously (less than 0.8). The behavior of back-off time manipulation of the malicious node will affect its idle time trust value, number of retransmissions trust value and physical layer trust value, so its trustworthiness will decrease to about 0.78. The selective forwarding attack will reduce the packet forward trust value of the malicious node. In the scenario of cross-layer attack, the parameters of both MAC layer and network layer will be affected, so the trust value of the malicious node will decrease markedly. The sinkhole attack will affect the hop count trust value, packet forward trust value, physical trust value of the malicious node because the malicious node will drop the Route Request (RREQ) packet, send the RREP packet with small hop count. Actually, the trust value of the malicious node is closely related to the selection of the attack parameters. In the cross-layer attack, in order to conceive its attack behavior, the malicious node will reduce the attack strength at the MAC layer and network layer, so in the simulation, the trust value of the malicious node in the cross-layer attack is slightly higher than that in the sinkhole attack.

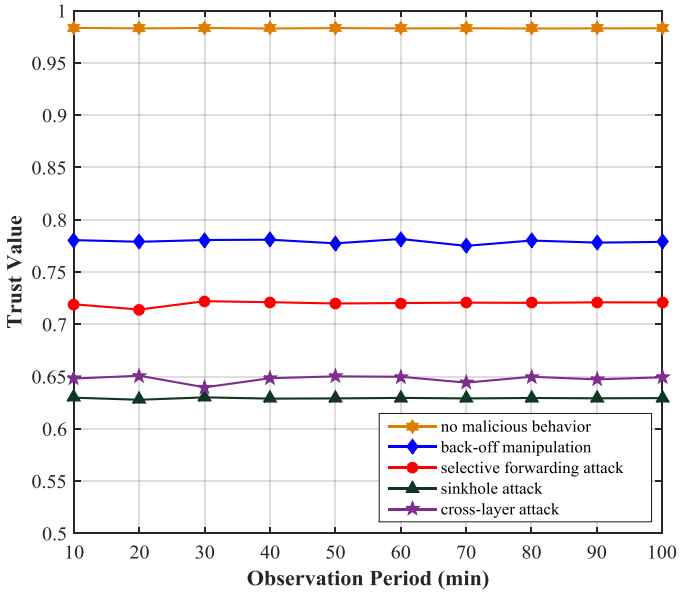

Figure 4. Trust values under different kinds of attacks.

To get the detection threshold, we simulate false positive and false negative probabilities of our scheme with different thresholds under different types of attacks. We observe that the false positive probability curves under different attacks are similar except fluctuations within a small range. This is because the attacks have little influence on the trust values of unmalicious nodes but greater impact on those of malicious nodes. The intersection of false positive probability curve and false negative probability curve is the optimal trust threshold. Under the four attacks, we obtain an optimal detection threshold at which both false negative and false positive probabilities are minimized. As illustrated in Figure 5, the optimal detection threshold is about 0.83 at which both false positive and false negative probabilities are less than 0.05 for all types of attacks. We also obtain the false positive and negative probabilities according to Equations (22) and (23). Figure 6 shows the theoretical results are consistent with the simulation results.

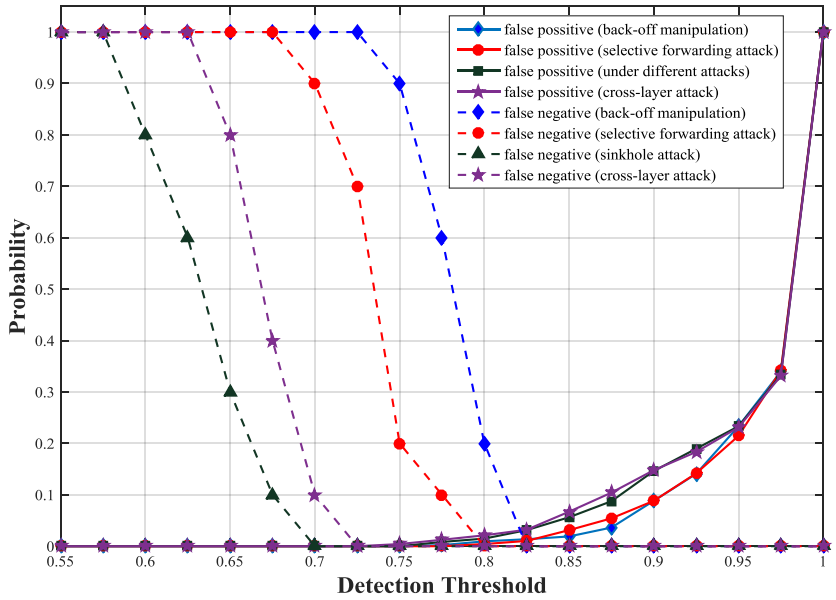

Figure 5. False positive and negative probabilities with different thresholds using simulation.

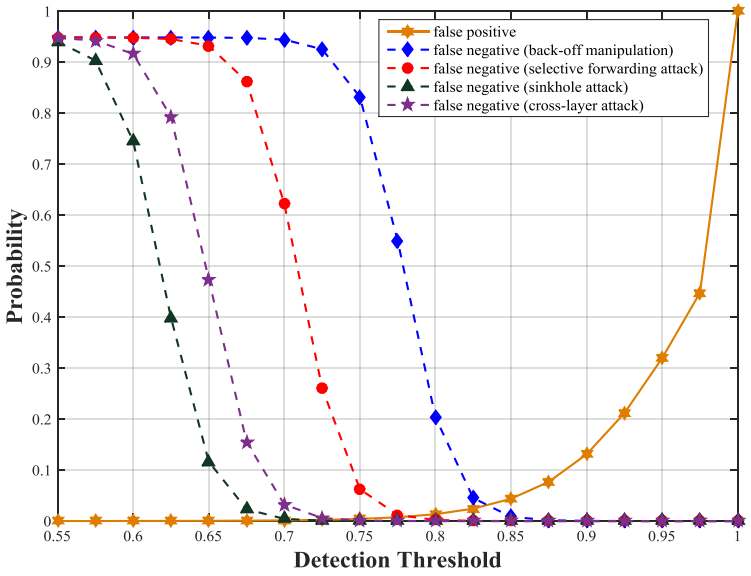

Figure 6. False positive and negative probabilities with different thresholds using formula calculation.

We analyze the influence of the proportion of malicious nodes on the detection probability using the optimal threshold 0.83, as illustrated in Figure 7. We observe that the detection probability of sinkhole attack is the highest among the four types of attacks and the detection probability of back-off manipulation attack is the lowest because in the simulation sinkhole attack influences the trust value of the malicious node strongly and back-off manipulation attack has the minimal impact on it. If the proportion of malicious nodes is less than 5%, the detection probability will be more than 97%. If the proportion of malicious nodes is greater than 5%, the detection probability will decrease obviously, because with the increase of the number of malicious nodes, the trust value of the unmalicious node is closer to that of the malicious node, and then it is difficult to distinguish between the unmalicious node and the malicious node.

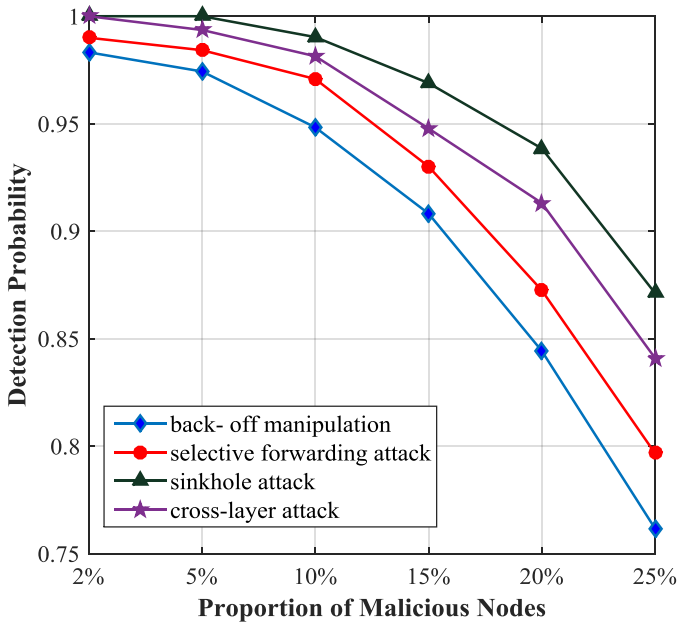

Figure 7. Detection probability with different proportion of malicious nodes.

Figure 8 describes the relationship between the false positive probability and the proportion of malicious nodes. If the proportion of malicious nodes is less than 5%, the false positive probability will be less than 0.05. It increases rapidly with the increase of the proportion of malicious nodes.

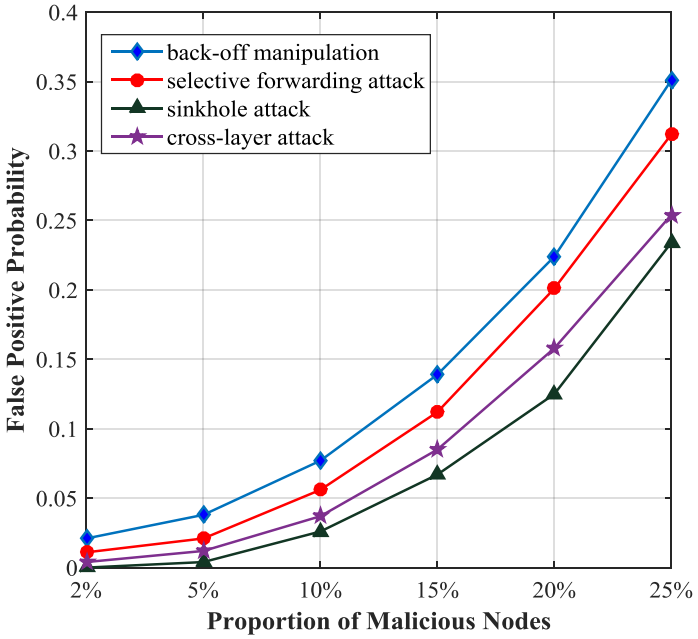

Figure 8. False positive probability with different proportion of malicious nodes.

We compare the detection probability of our scheme with that of the NBBTE [23]. As shown in Figure 9, the detection probability of the selective forwarding attack and sinkhole attack have been improved by more than 10% and that of cross-layer attack has been improved by more than 20% as the proportion of malicious nodes is 2%, because many key parameters of multiple protocol layers are monitored and the trust values of SNs are calculated more accurately in our scheme. In NBBTE, the back-off manipulation attack can hardly be detected, because NBBTE only focuses on the node behaviors at the network layer, but ignores the malicious behaviors at the MAC layer, so it is not effective to detect the attacks at the MAC layer.

Figure 10 shows that the false positive probability of NBBTE is higher than that of our scheme. Because we use the deviations of protocol parameters instead of the variations of node behaviors as NBBTE does for detecting malicious node, the reduction of the trust value caused by the normal change of the network can be avoided in our scheme. In NBBTE, the false positive probability curves under different attacks are very similar because the malicious behaviors have little influence on the trust values of normal nodes according to their algorithm.

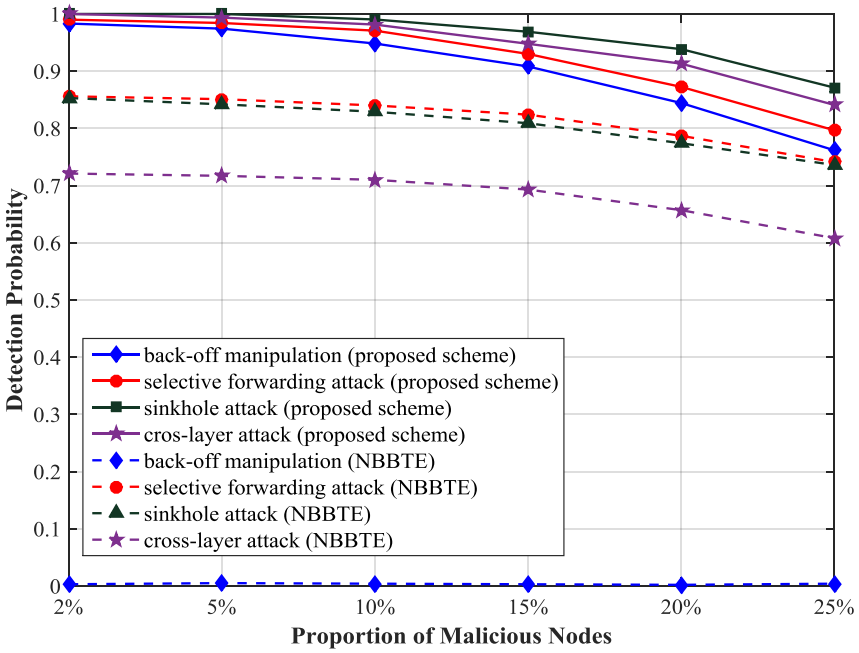

Figure 9. Comparison of the detection probability. NBBTE = Node Behavioral strategies Banding belief theory of the Trust Evaluation algorithm.

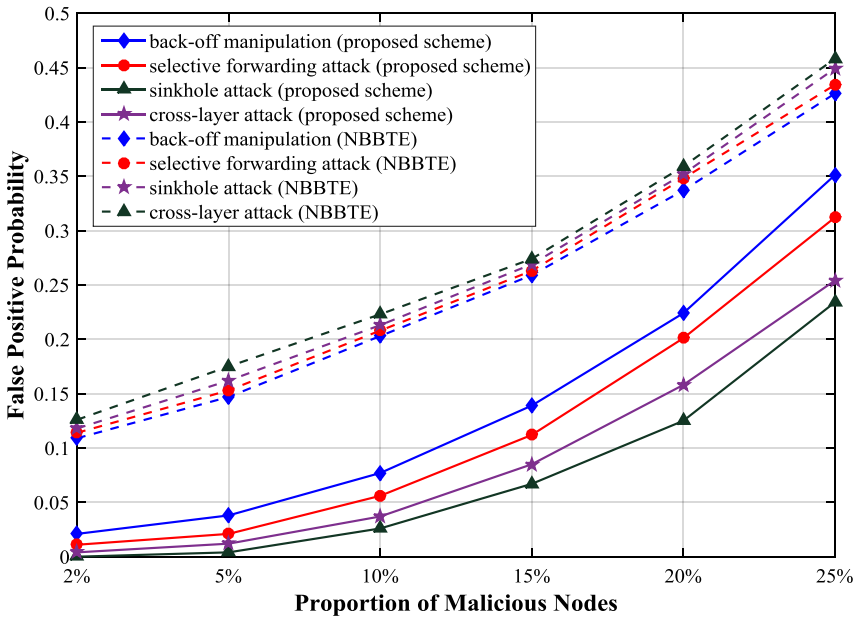

Figure 10. Comparison of the false positive probability.

4.3. Analysis of Communication Overhead

In our scheme, each sensor node monitors the key parameters of its neighboring nodes at each protocol layer and transmits the trust values toward the monitored nodes to its CH, so the communication overhead of our scheme mainly comes from the packets transmitted from SNs to the CH. As a result, the communication overhead of our scheme is related to the hop count from SNs to the CH. In NBBTE, it includes direct evaluation and indirect evaluation. In the direct evaluation, the monitoring node collects the key parameters of the monitored node and calculates the trust factors of the corresponding parameters. There is a factor of availability which evaluates the availability

of the neighboring nodes. To obtain this factor, the monitoring node needs to transmit a HELLO packet and its neighboring nodes should reply to it with ACK-HELLO packets. In the indirect evaluation, the neighboring nodes of the monitored node will transmit their trust evaluation results towards the monitored node to the monitoring node as the indirect recommendation values. Thus, the communication overhead of NBBTE includes two parts, the HELLO packets for the factor of availability and the indirect trust evaluation from the recommendation nodes, which are related to the average number of neighboring nodes in the network.

In our scheme, all SNs transmit their evaluation results to their CH once in an observation period. Assuming there are n SNs in a cluster and the average hop count to the CH is N_h, the communication overhead of our scheme, CO_p in an observation period can be expressed as $CO_p = nN_h$. As for the NBBTE, all SNs broadcast HELLO packets and reply to the HELLO packets of their neighboring nodes once in an observation period. Meanwhile, the neighboring nodes of each SN will send their recommendation values once in an observation period. Assuming the number of SNs is n, and the average number of neighboring nodes of a SN is N_a, the communication overhead of NBBTE, CO_N in an observation period can be denoted by $CO_N = n(2N_a + 1)$. We then analyze the communication overhead of the two schemes quantitatively in a network with 50 SNs and 1 CH under the circumstance that the two schemes have the same observation period Δt.

Figure 11 shows the comparison of the communication overhead of the two schemes under different number of neighbor nodes in the case that the average hop count N_h in the network is 6. The communication overhead of the proposed scheme is not related to the number of neighboring nodes. If the average number of neighboring nodes is less than or equal to 2, the communication overhead of the NBBTE is less than that of our scheme. However, if the average number of neighboring nodes is greater than or equal to 3, the communication overhead of the NBBTE will be greater than that of our scheme.

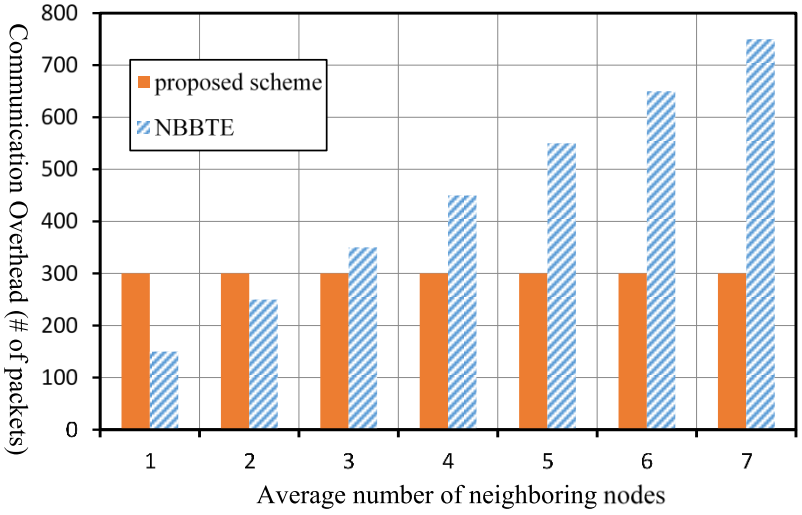

Figure 11. Comparison of the communication overhead under different number of neighboring nodes.

Figure 12 shows the comparison of the communication overhead of the two schemes under different average of hop count in the case that the average number of neighboring nodes N_a in the network is 3. The communication overhead of our scheme will increase with increasing of the average hop count to the CH. If the average hop count to the CH is greater than 7, the communication overhead of our scheme will be greater than that of the NBBTE.

As described above, if the number of average neighboring nodes in a network is relatively large, the communication overhead of the NBBTE is greater than that of our scheme. If the hop count to the CH is relatively large, the communication overhead of our scheme is greater than that of the NBBTE. As a result, our scheme is more applicable to the network with less hop count. Moreover,

from the angle of computation complexity, the calculation of trust value and the decision approach in the NBBTE are more complex than those in our scheme.

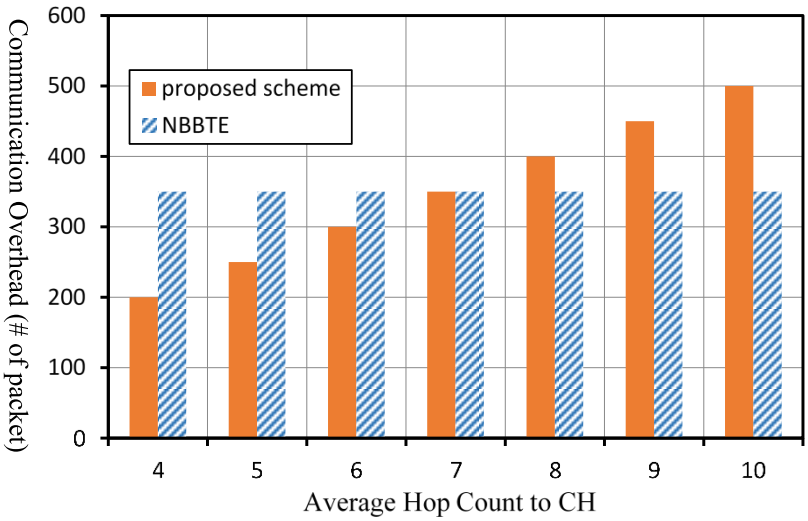

Figure 12. Comparison of the communication overhead under different average of hop count to the CH.

5. Conclusions

Wireless sensor networks are vulnerable to variety of attacks at different protocol layers. In the existing trust-based intrusion detection schemes, there is no unified standard to select trust factors, and cross-layer attacks are seldom considered. In order to identify malicious nodes more efficiently, we have proposed a protocol layer trust-based intrusion detection scheme for WSNs. In our scheme, the key parameters of different protocol layers are monitored and the trust values of sensor nodes can be calculated according to the deviations of parameters. By comparing the trust value with a predefined threshold, we can decide whether the sensor node is compromised or not. It can describe the trust values of sensor nodes more accurately by considering the deviations of parameters of multiple layers, hence our proposed scheme is effective for detecting cross-layer attacks. We utilized the t-distribution and simulation to analyze the detection probability and false positive probability of our scheme. The results indicate that there exists an optimal trust threshold at which both false positive and false negative probabilities are minimized. Our proposed scheme outperforms the NBBTE scheme in terms of the detection probability and false positive probability. The weakness of our scheme is the communication overhead will increase with the increasing of the hop count to the CH. Our scheme is extendable, the selection of the trust factors at different protocol layers can be adjusted according to the requirements of a system, and it is applicable to both clustered WSNs and flat WSNs. As for future works, we will analyze the attacks initiated at the transport layer and application layer, as well as MAC-Transport cross-layer attack, Network-Application cross-layer attack and their influence on the protocol parameters to further optimize our scheme. In addition, we will perform experiments to test the performance of our scheme on a real WSN testbed to assess its real-life performance.

Author Contributions: All authors made significant contributions to this paper. Jian Wang designed the scheme and wrote the paper. Jiang Shuai performed the simulation and analyzed the performance of the scheme. Abraham O. Fapojuwo provided valuable insights for the scheme and revised the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wood, A.D.; Stankvic, J.A. Denial of service in sensor networks. *IEEE Comput.* **2002**, *35*, 54–62. [CrossRef]

2. Wang, X.; Gu, W.; Schosek, K.; Chellappan, S.; Xuan, D. *Sensor Network Configuration under Physical Attacks*; Technical Report: OSU-CISRC-7/04-TR45; Department of Computer Science and Engineering, Ohio State University: Columbus, OH, USA, 2004.
3. Xu, W.; Ma, K.; Trappe, W.; Zhang, Y. Jamming sensor networks: Attack and defense strategies. *IEEE Netw.* **2006**, *20*, 41–47.
4. Raymond, D.R.; Marchany, R.C.; Brownfield, M.I.; Midkiff, S.F. Effects of denial-of sleep attacks on wireless sensor network MAC protocols. *IEEE Trans. Veh. Technol.* **2009**, *58*, 367–380. [CrossRef]
5. Sokullu, R.; Dagdeviren, O.; Korkmaz, I. On the IEEE 802.15.4 MAC layer attacks: GTS attack. In Proceedings of the Second International Conference on Sensor Technologies and Applications (SENSORCOMM), Cap Esterel, France, 25–31 August 2008; pp. 673–678.
6. Radosavac, S.; Crdenas, A.A.; Baras, J.S.; Moustakides, G.V. Detecting IEEE 802.11 MAC layer misbehavior in Ad Hoc networks: Robust strategies against individual and colluding attackers. *J. Comput. Secur.* **2007**, *15*, 103–128. [CrossRef]
7. Karlof, C.; Wagner, D. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Netw. J.* **2003**, *1*, 293–315. [CrossRef]
8. Hu, Y.C.; Perrig, A.; Johnson, D.B. Packet Leashes: A defense against wormhole attacks in wireless networks. In Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM), San Francisco, CA, USA, 1–3 April 2003; Volume 3, pp. 1976–1986.
9. Al-Shurman, M.; Yoo, S.M.; Park, S. Black hole attack in mobile ad hoc networks. In Proceedings of the 42nd Annual ACM Southeast Regional Conference (ACM-SE'42), Huntsville, Alabama, AL, USA, 2–3 April 2004; Volume 33, pp. 96–97.
10. Newsome, J.; Shi, E.; Song, D.; Perrig, A. The sybil attack in sensor networks: Analysis & defenses. In Proceedings of the third International Symposium on Information Processing in Sensor Networks ACM (IPSN), Berkeley, CA, USA, 26–27 April 2004; pp. 259–268.
11. Radosavac, S.; Benammar, N.; Baras, J.S. Cross-layer attacks in wireless ad hoc networks. In Proceedings of the 38th Annual conference on Information Science and Systems (CISS), Princeton, NJ, USA, 17–19 March 2004; pp. 1266–1271.
12. Bian, K.; Park, J.M.; Chen, R. Stasis Trap: Cross-layer stealthy attack in wireless ad hoc networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, CA, USA, 27 November–1 December 2006; pp. 1–5.
13. Nagireddygari, D.; Thomas, J. MAC-TCP cross-layer attack and its defense in cognitive radio networks. In Proceedings of the 10th ACM International Symposium on QoS and Security for Wireless and Mobile Networks, Montreal, QC, Canada, 21–22 September 2014; pp. 71–78.
14. Depren, O.; Topallar, M.; Anarim, E.; Ciliz, M.K. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Syst. Appl.* **2005**, *29*, 713–722. [CrossRef]
15. Han, Z.J.; Wang, R.C. Intrusion detection for wireless sensor network based on traffic prediction model. *Phys. Procedia* **2012**, *25*, 2072–2080.
16. Palpanas, T.; Papadopoulos, D.; Kalogeraki, V.; Gunopulos, D. Distributed deviation detection in sensor networks. *ACM Sigmod Rec.* **2003**, *32*, 77–82. [CrossRef]
17. Moshtaghi, M.; Havens, T.C.; Bezdek, J.C.; Park, L.; Leckie, C.; Rajasegarar, S.; Keller, J.M.; Palaniswami, M. Clustering ellipses for anomaly detection. *Pattern Recognit.* **2011**, *44*, 55–69. [CrossRef]
18. Javidi, M.M.; Aliahmadipour, L. Game theory approaches for improving intrusion detection in MANETs. *Sci. Res. Essays* **2011**, *6*, 6535–6539.
19. Shen, S.; Yue, G.; Cao, Q.; Yu, F. A Survey of game theory in wireless sensor networks security. *J. Netw.* **2011**, *6*, 521–532. [CrossRef]
20. Mohi, M.; Movaghar, A.; Zadeh, P.M. A bayesian game approach for preventing DoS attacks in wireless sensor networks. In Proceedings of the WRI International Conference on Communications and Mobile Computing, Kunming, China, 6–8 January 2009; pp. 1695–1699.
21. Lira, A.T.H.; Lau, H.K.; Timmis, J.; Bate, I. Immune-inspired self healing in wireless sensor networks. In Proceedings of the 11th International Conference on Artificial Immune Systems, Taormina, Italy, 28–31 August 2012; Volume 7597, pp. 42–56.
22. Ishmanov, F.; Malik, S.A.; Kim, S.W.; Begalov, B. Trust management system in wireless sensor networks: Design considerations and research challenges. *Trans. Emerg. Telecomun. Technol.* **2013**. [CrossRef]

23. Feng, R.; Xu, X.; Zhou, X.; Wan, J. A trust evaluation algorithm for wireless sensor networks based on node behaviors and d-s evidence theory. *Sensors* **2011**, *11*, 1345–1360. [CrossRef] [PubMed]
24. Wu, R.; Deng, X.; Lu, R.; Shen, X. Trust-based anomaly detection in wireless sensor networks. In Proceedings of the 1st IEEE International Conference on Communications in China: Communications Theory and Security, Beijing, China, 15–17 August 2012; pp. 203–207.
25. Shao, N.; Zhou, Z.; Sun, Z. A Lightweight and Dependable Trust Model for Clustered Wireless Sensor Networks. In Proceedings of the International Conference on Cloud Computing and Security, Nanjing, China, 13–15 August 2015; pp. 157–168.
26. Ganeriwal, S.; Balzano, L.K.; Srivastava, M.B. Reputation based framework for high integrity sensor networks. In Proceedings of the 2nd ACM Workshop Security Ad Hoc Sensor Network, Washington, DC, USA, 25–29 October 2004; pp. 66–77.
27. Luo, W.; Ma, W.; Gao, Q. A dynamic trust management system for wireless sensor networks. *Sec. Commun. Netw.* **2016**, *9*, 613–621. [CrossRef]
28. Atakli, I.M.; Hu, H.; Chen, Y.; Ku, W.S.; Su, Z. Malicious node detection in wireless sensor networks using weighted trust evaluation. In Proceedings of the Symposium on Simulation of Systems Security, Ottawa, ON, Canada, 14–17 April 2008; pp. 836–843.
29. Shaikh, R.A.; Jameel, H.; d'Auriol, B.J.; Lee, H.; Lee, S. Group-based trust management scheme for clustered wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2009**, *20*, 1698–1712. [CrossRef]
30. Yao, Z.; Kim, D.; Doh, Y. PLUS: Parameterized and localized trust management scheme for sensor networks security. In Proceedings of the IEEE International Conference on Mobile Ad hoc & Sensor System, Vancouver, BC, Canada, 9–12 October 2006; pp. 437–446.
31. Li, X.; Zhou, F.; Du, J. A lightweight and dependable trust system for clustered wireless sensor networks. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 924–935. [CrossRef]
32. Jiang, J.; Han, G.; Wang, F.; Shu, L.; Guizani, M. An efficient distributed trust model for wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1228–1237. [CrossRef]
33. Ishmanov, F.; Kim, S.W.; Nam, S.Y. A robust trust establishment scheme for wireless sensor networks. *Sensors* **2015**, *15*, 7040–7061. [CrossRef] [PubMed]
34. Bao, F.; Chen, I.R.; Chang, M.; Chao, J.H. Trust-based intrusion detection in wireless sensor networks. In Proceedings of the IEEE International Conference on Communications, Kyoto, Japan, 5–9 June 2011; pp. 1–6.
35. Bao, F.; Chen, I.R.; Chang, M.; Chao, J.H. Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection. *IEEE Trans. Netw. Serv. Manag.* **2012**, *9*, 169–183. [CrossRef]
36. Zhang, T.; Yan, L.; Yang, Y. Trust evaluation method for clustered wireless sensor networks based on cloud model. *Wirel. Netw.* **2016**. [CrossRef]
37. Rajeshkumar, G.; Valluvan, K.R. An Energy Aware Trust Based Intrusion Detection System with Adaptive Acknowledgement for wireless sensor network. *Wirel. Pers. Commun.* **2016**. [CrossRef]
38. Wang, J.; Fapojuwo, A.O.; Zhang, C.; Tan, H. UML Modeling of Cross-Layer Attack in Wireless Sensor Networks. In Proceedings of the 2nd International Conferenceon on safety and security in IoT, Paris, France, 26–27 October 2016; pp. 104–115.
39. Woo, A.; Tong, T.; Culler, D. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In Proceedings of the 1st International Conference on Embedded networked sensor systems (SenSys), Los Angeles, CA, USA, 5–7 November 2003; pp. 14–27.
40. Tiny, O.S. Webpage. Available online: http://www.tinyos.net/ (accessed on 25 March 2017).
41. Wang, J.; Yang, L.; Zhao, Y. Research of wireless sensor networks cross-layer energy optimization based on link quality. In Proceedings of the 3rd International Conference on Measuring Technology and Mechatronics Automation, Shanghai, China, 6–7 January 2011; pp. 1092–1094.
42. Abdullah, M.I.; Rahman, M.M.; Roy, M.C. Detecting sinkhole attacks in wireless sensor network using hop count. *Int. J. Comput. Netw. Inf. Secur.* **2015**, *3*, 50–56.