Indonesian Journal of Electrical Engineering and Computer Science
Vol. 5, No. 1, January 2017, pp. 147 ∼ 158
DOI: 10.11591/ijeecs.v5.i1.pp147-158

Attacks and Secure Geographic Routing in Wireless Sensor Networks

Yassine Sabri* and Najib El Kamoun
Chouaib Doukkali University, B.P: 20 . El Jadida Morocco
*Corresponding author, e-mail: sabriyassino@gmail.com

Abstract

Due to open network nature of wireless sensor networks make them highly vulnerable to a variety of security attacks and easy target for adversaries, which may capture these nodes, analyze and easily insert fake route information. Wireless sensor network is an emerging, cost effective and unsupervised solution for collecting this information from the physical world and sending this information back to centralized authority for further processing. GRPW (Geographic Routing in connected wireless sensor networks based on Multiple Sinks) is one of the basic routing protocols used for Supporting Mobile Sinks in Wireless Sensor Networks . GRPW , a geographical routing protocol for wireless sensor networks , is based on an architecture partitioned by logical levels, on the other hand based on a multipoint relaying flooding technique to reduce the number of topology broadcast. GRPW-MuS uses periodic HELLO packets to neighbor detection. As introduced in Reference [1, 2], the wormhole attack can form a serious threat in wireless sensor networks, especially against many wireless sensor networks routing protocols and location-based wireless security systems. Here, a trust model to handle this attack in GRPW is provided called GRPW-MuS-s . Using OMNET++ simulation and the MiXiM framework, results show that GRPW-MuS-s protocol only has very small false positives for wormhole detection during the neighbor discovery process (less than GRPW). The average energy usage at each node for GRPW-MuS-s protocol during the neighbor discovery and route discovery is very low than GRPW-MuS, which is much lower than the available energy at each node. The cost analysis shows that GRPW-MuS-s protocol only needs small memory usage at each node , which is suitable for the sensor network.

Keywords: *Wireless Sensor Network (WSN), Routing, Security, Wormhole attack*

1. Introduction

Many sensor network applications, such as emergency response operations in a disaster environment or battlefield monitoring, that run in untrustworthy environments, require secure communication and routing [3–5] to safeguard against different types of attacks. The attacks such as blackhole, wormhole, misdirection, and replay [6, 7] can cause an existing route to be broken or a new route to be prevented from being established [8, 9]. There are several examples of attacks against routing in sensor networks; a routing packet could be captured and the information in the packet could be tampered with, or the adversary might insert a spurious message in the sensor network. Traditional security protocols are designed for resource rich machines to support large computation and are not applicable to sensor networks due to resource limitations, ad hoc nature, and intermittent connectivity. Many sensor network routing protocols have been proposed, but very few of them have been designed with secure routing as a goal. Secure routing protocols in sensor networks present challenges, which do not exist in traditional networks, such as no centrally administered routers, low power, and small memory nodes.

A wormhole is a tunnel which connects two remote nodes. In a wormhole attack [10], an attacker receives packets at one location in the network, tunnels them to a remote location in the network, and then replays them into the network from that location. A wormhole attack can be easily executed against routing in sensor networks because it does not need to physically compromise any sensor node. Thus, a wormhole attack poses a serious threat against routing in

Received November 12, 2016; Revised December 19, 2016; Accepted December 30, 2016

the sensor network as most of routing protocols do not have any mechanism to defend against it. A wormhole attack can cause the sensor nodes in the target area to build a route through an attacker which can later tamper with the data messages, or selectively forward data messages. However, most of the researchers proposed solutions against a wormhole attack during the neighbor discovery process with the use of some special hardware [11–13]. Moreover, their approach did not focus on how to build a secure route against the wormhole attack without any additional special hardware, such as a directional antenna, GPS, and a synchronized clock.

In a multihop wireless ad hoc Network, mobile nodes cooperate to form a Network without using any infrastructure such as access points or base stations. Instead, the mobile nodes forward packets for each other, allowing communication among nodes outside wireless transmission range. The nodes' mobility and the fundamentally limited capacity of the wireless medium, together with wireless transmission effects such as attenuation, multipath propagation, and interference, combine to create significant challenges for routing protocols operating in an ad hoc network. Several routing protocols for wireless sensor networks have been developed . GRPW-MuS was proposed in [14], which belongs to the geographical for wireless sensor networks class of routing protocols. GRPW-MuS is an optimization of the classical geographical algorithm tailored to the requirements of a mobile wireless . The key concept used in the protocol is multlevels relays (MLRs). MLRs are nodes selected in charge of forwarding broadcast messages during the flooding process in each logical level. This technique substantially reduces the message overhead as compared with a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message. So this protocol is particularly suitable for large and dense Network. In Reference[15], attacks on WSNs protocols generally fall into one of two following categories: routing-disruption attacks and resource consumption attacks. Wormhole attack is classified into routing-disruption attacks. In the wormhole attack, an attacker records packets (or bits) at one location in the Network, tunnels them to another location, and relays them there. Due to the nature of wireless transmission, the attacker can create a wormhole even for packets not addressed to itself, since it can overhear them in wireless transmission and tunnel them to the colluding attacker at the opposite end of the wormhole.

The GRPW-MuS's neighbor discovery mechanisms rely heavily on the reception of HELLO packets to neighbor detection, so it is extremely vulnerable to this attack. When an attacker tunnels through a wormhole to a colluding attacker near node B all HELLO packets transmitted by node A, and likewise tunnels back to the first attacker all HELLO packets transmitted by B, then A and B will believe that they are neighbors, which would cause the routing protocol to fail to find routes when they are not actually neighbors. Furthermore, the attacker is invisible at higher layers, unlike a malicious node in a routing protocol, which can often easily be named, the presence of the wormhole and the two colluding attackers at either endpoint of the wormhole are not visible in the route.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 describes the problem statement. Section 4 provides an overview of GRPW-MuS approach. Section 5 gives a detailed description of GRPW-MuS-S approach. Section 6 gives cost analysis. Section 7 gives performance evaluations, and Section 8 concludes the paper.

2. RELATED WORK AND BACKGROUND

The important approach for preventing wormhole attacks is presented in References [16]. The main idea is that by authenticating either an extremely precise timestamp or location information combined with a loose timestamp, a receiver can determine if the packet has traversed an unrealistic distance for the specific network technology used. Temporal leashes rely on extremely precise time synchronization and timestamps in each packet. But to construct a temporal leash, all nodes must have tightly synchronized clocks, which in fact are not easy to achieve in MANET. Geographical leashes rely on all nodes knowing its own location and having loosely synchronized clock. In that paper, the authors also point out that in some circumstances, bounding the distance between the sender and receiver, cannot prevent wormhole attacks. Another method of preventing wormhole tacks is known as RF watermarking- -, which authenticates a wireless transmission by modulating the RF waveform in a way known only to authorize node. But if the radio band in which communications are taking place is known, then an attacker can attempt to tunnel the entire signal from one location to another. Some authors also propose using intrusion detection to handle the wormhole attack, but intrusion detection is difficult to isolate the attacker in a software-only approach.

In [17] presented a general mechanism, called packet leashes, for detecting and thus defending against wormhole attacks in wireless networks. They presented two types of packet leashes: geographic leashes and temporal leashes. A geographical leash ensures that the recipient of the packet is within a certain distance from the sender. To construct a geographical leash, each node must know its own location, and all nodes must have loosely synchronized clocks. A temporal leash ensures that the packet has an upper bound on its lifetime, which restricts the maximum travel distance since the packet can travel as fast as the speed of light. To construct a temporal leash, all nodes must have tightly synchronized clocks. The disadvantage of using packet leashes is that they require either location information for each node or need tight clock synchronization between the nodes.

In [18] have presented a solution that uses directional antennas by mobile nodes to defend against wormholes. Their assumption is that if there is no wormhole attack and if one node sends packets in a given direction, then its neighbor will get that packet from the opposite direction. The neighboring nodes examine the directions of the received signals from each other with a shared witness. Only when the directions of both pairs match, the neighboring relation is confirmed. The disadvantage is that each node is to be equipped with the special hardware called directional antenna, which is not always possible.

In [19] proposed a graph theoretic model for characterizing a wormhole attack and derived the necessary and sufficient conditions for any candidate solution to prevent wormholes. In this approach, a small fraction of the nodes needs to be equipped with a GPS receiver. In [20] proposed a mechanism, MDSVOW, to detect wormholes in a sensor network. MDS-VOW detects a wormhole by visualizing the anomalies introduced by an attack. The anomalies, which are caused by the fake connections through the wormhole, bend the reconstructed surface to pull the sensors that are far away to each other. By detecting this bending feature, the wormhole is located and fake connections are identified. The disadvantage is that the message overhead is high because all of the sensors need to send their neighbor lists to the base station. In [21], the authors proposed two mechanisms based on hypothesis testing for detecting wormholes in wireless sensor networks. The first mechanism, called the neighbor number test (NNT), detects increases in the number of neighbors of the sensors due to new links created by the wormhole in the network. The second mechanism, called the all distances test (ADT), detects decreases in the lengths of the shortest paths between all pairs of sensors, which are due to the shortcut links created by a wormhole in the network. Both mechanisms assume that the sensors send their neighbor lists to the base station and the base station runs the algorithms on the network topology. The disadvantages are (1) the message overhead is high because all of the sensors need to send their neighbor lists to the base station and (2) the mechanisms can only detect the presence of a wormhole, but they do not pinpoint its exact location.

3. Problem statement

Recall, in a wormhole attack, an attacker receives packets at one location in the network, tunnels them to another location, and retransmits them there into the network. In the basic route discovery process, the base station starts the route discovery by broadcasting a routing beacon. Each node which receives the routing beacon records the base station's identity as its parent. Then, it rebroadcasts the routing beacon. The algorithm continues recursively with each node marking the first node from whom it hears a route beacon to be its parent. The basic route discovery process fails if an attacker receives the routing beacon at one point in the network, tunnels it to another point in the network, and then replays it into the network from that point. Since the routing beacon tunneled by the wormhole reaches the targeted area considerably faster than it normally would have through the multi-hop routing, the nodes near the endpoint of the wormhole will create a large routing sub-tree in the targeted area with themselves as the root. For exemple, the attacker tunnels the routing beacon from M_1 to M_2. The nodes in the target area build the route through the wormhole located between M_1 and M_2. All the traffic in the targeted area will be channeled through the wormhole. If an attacker performs this tunneling honestly and reliably, no harm is done; the attacker actually provides a useful service in connecting the network more efficiently [22]. However, the wormhole puts the attacker in a very powerful position relative to other nodes in the network. The attacker discards rather than forwarding all the data packets. Thereby, it creates a permanent denial-of-service attack, where the base station cannot receive any information from the target area. Also, the attacker can exploit the wormhole to selectively discard or modify certain data packets. System assumption. We assume that the sensor nodes after deployment are not movable. Each sensor node has the same energy at the start. It has a unique identity (ID) and an initial key KI and the random function f. We assume that the initial key KI is stored in the memory, which can be erased completely [23]. The sensor nodes communicate using RF (radio frequency), so broadcast is the fundamental communication primitive [24]. Two nodes within each other's transmission range are called one-hop neighbors. We assume that communication channels are bidirectional [24], i.e. if a node y can receive a message from z, then it can also send a message to z. We assume that the channel, based on MAC protocols [25], between the sensor nodes is reliable. That is, the signals sent from different sensor nodes across the same channel do not collide.

4. Security scheme

We use an adaptation of the trust model [26] configured by Marsh for use in pure ad hoe Networks. Marsh's model computes situational trust in agents based upon the general trust in the trustor and in the importance and utility of the situation in which an agent finds itself. General trust is basically the trust that one entity assigns another entity based upon all previous transactions in all situations. In our model each node have a trust evaluator which gathers data from the neighbor's events in all states, filters it, assigns weights to each event and computes different trust levels based upon them. The trust evaluator has three functions: trust derivation, quantification, and computation. At first, in GRPW-Mus the trust can come from the information about the successful transmission of any packet that is relayed by the neighboring node, such as some acknowledgments. Second, the neighboring node's HELLO packet received on schedule can also conduce to the trust. These events can be categorized into data and control packet types, and in each event there are two states: success and fail, which record the number of successful events and failed events respectively. In trust quantification process, we represent trust from -1 to 1 signifying a continuous range from complete distrust to complete trust. Trust computation involves an assignment of weights to the event that were monitored and quantified. We use the continuous range from 0 to 1 for representing the significance of a certain event from unimportant to most important. The higher weights represent the event more important. We define the trust T to the neighboring node y by the node x, and it is given by the following equation:

$$T_x(y) = \sum_{i=1}^{n} [W_x(i) \times T_x(i)] \tag{1}$$

where $W_x(i)$ is the weight of the *i*th trust category to x and $T_x(i)$ is the situational trust of x in the *i*th trust category. The n represents the number of category. From above equation, we can get the following equations :

$$C_h = \frac{H_S - H_F}{H_S + H_F} for H_S + H_F \neq 0 \; else \; C_h = 0 \tag{2}$$

Negative values represent that more failed events occur than successes. Hence, a value of -1 represents complete distrust, a value of 0 implies a non-contributing event and a value of $+1$ means absolute trust in a particular event. Now the node x can get the whole trust T to the neighboring node y.

$$T_x(y) = W_x(C_h) \times T_x(C_h) + W_x(C_d) \times T_x(C_d) \tag{3}$$

5. GRPW-MuS review

In this section we will focus on introducing the GRPW-MuS Routing approach as this is the foundation for our work. For a more elaborate description to GRPW-Mus please refer to [14].

GRPW-Mus Is a geographic routing protocol for wireless sensor networks for multiple sink, Based on a partitioned topology in circular logic levels ,each node can get its own location information either by GPS or other location services. The routing of data is inspired by the principle of water flow in a washbasin by creating the virtual logic levels as described in the figure 1 and 2 . After this logical network reconstruction ,each sink establishes its area based on the sink DS position. The routing of captured data be performed within each zone belonging to each node using the GRPW-Mus method for each Area Sink .

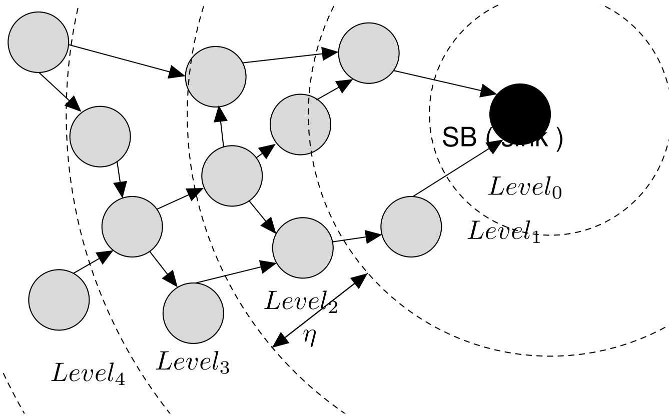

Figure 1. Illustration of GRPW-MuS routing network levels

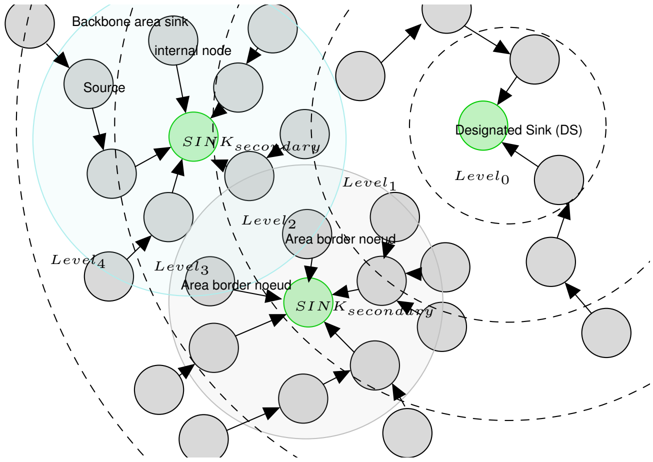

Figure 2. Illustration of GRPW-MuS routing network levels

The procedure of GRPW-MuS is as follows. Every node broadcasts HELLO messages that contain one-hop neighbor information periodically. The TTL of HELLO messages is 1, so they should not forwarded by its neighbors. With the aid of HELLO messages, every node obtains local topology information. A node (also called DS) chooses a subset of its neighbors to act as multipoint relaying nodes for it is based on the local Level topology information, which Level specified in the periodic HELLO messages later. DS nodes perform two tasks:

1. when the Sink sends or forwards a broadcast packet, only its DS nodes among all its neighbors forward the packet
2. the DS nodes periodically broadcast its selector list . Thus every node in the each level knows through which DS nodes every other node could be reached.

With each level's topology information stored and updated at every node, a shortest path from one node to every other node could be computed with GRPW-Mus algorithm, which goes along a series of DS node.

6. Extension to GRPW-MuS

The framework of extension to GRPW-Mus is shown in Fig.3

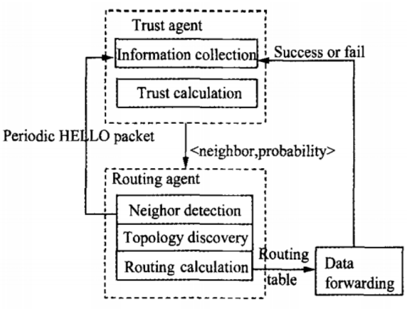

Figure 3. Framework of extension to GRPW-MuS

When the node receives a new sender's HELLO message, it will make two new records ¡node, positive, negative, event¿ , to record separately the event of this sender's HELLO message's coming in time or not, and the event of data forwarding successfully or not. Then in information collection there are two tables to record every possible neighbor's events. These tables are the inputs of trust calculation. By trust calculation, every possible neighbor will get a value which represents the probability of the neighbor relationship. The tuples neighbor, probability ¿ will be recorded in Neighbor Set. Some GRPW-MuS information repositories and packets' format should be modified. When the node broadcasts the HELLO message, it contains its neighbor information including the recommendation about the probability of neighbor relationships. From receiving others HELLO messages, every node obtains local topology information. When choosing MPR nodes, the node will take the nodCs recommendation as an important factor. When nodes exchange the Hello messages which contain the information about the neighbor relationship' s probability, every node would get global topology information which can construct a weighted directed graph. The weight on the edge represents the evaluation of edge start point on the link existence between itself and the end point.Then from the weighted directed graph of the global topology, we can use Dijkstra algorithm to calculate the routing table. In this process, the probability of the "being a neighbor" is considered as the weight.

7. Performance evaluations

For performance analysis, we have simulated GRPW-MS-S protocol using OMNET++ discrete event simulator [27]. As OMNET++ is not developed for the sensor network, a sensor network environment is created in OMNET++ with the installation of a MiXiM framework patch [28]. In this simulation, we simulate 1600 sensor nodes. The transmission range for each sensor node is $40m$. Transmit Power P_t is the power with which the signal is transmitted. The Transmit

Power P_t decides the transmission range for the sensor node. Transmit Power (txPower) is the power consumed by the transceiver to transmit a data packet. Receive Power (rxPower) is the power consumed to receive a data packet.
we can see that there are some false positives, which means that some nodes are mistakenly detected to be connected by the wormhole since they are actually close nodes. In this section, we simulate the false positives under different deployments and different thresholds used . The purpose of this simulation is to control the false positives to the minimum. We design four different types of sensor deployment:

1. Random deployment within the grid (RandomGrid): The whole sensor deployment area is divided into grids with only one sensor node for each grid. The position of the sensor node in the grid is random.
2. Random deployment in the whole area (RandomArea): All sensor nodes are randomly deployed in the whole deployment area.
3. Normal distribution within the grid (NormalGrid): The whole sensor deployment area is divided into subareas, where each sub-area holds an equal number of sensor nodes. More specifically, let the total number of sensor nodes be N, the total number of divided grids be N_{grid}, then each grid contains (N/N_{grid}) sensor nodes. Within each grid, the sensor nodes are deployed using the normal distribution.
4. Normal distribution in the whole area (NormalArea): All sensor nodes are deployed in the whole deployment area following the normal distribution.

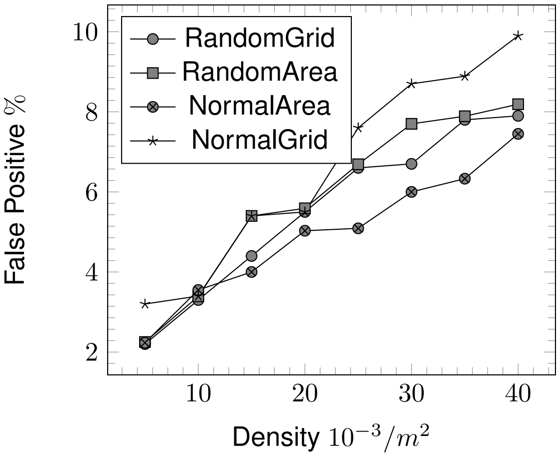

Figure 4. False positive vs. density ($Th = 4$).

Figs. 5 and 4 describe the relationship between the false positives and the density of the sensor network with the different types of sensor deployment under a specific threshold Th. We find that when the threshold Th is equal to or less than 6, all of the four deployments have false positives that are less than 10%. From Fig 4, we find that the NormalGrid has higher false positives than other deployments. Moreover, the false positives for NormalGrid deployment increases when the density of the sensor nodes increases. This is because in NormalGrid, when the density increases, each grid covers more nodes. Since nodes in each grid are deployed with the normal distribution, the nodes have more chance to become close nodes in each grid. This causes more false positives. We cannot see much difference in the false positives for the other deployments, which are RandomGrid, RandomArea, and NormalArea. Their false positives are low (less than 10%) if the threshold Th is below 12. Moreover, we find that the false positives are roughly the same when the density of the sensor network increases. This is because in RandomGrid/RandomArea, the nodes are randomly deployed. The nodes could be closer but they are still

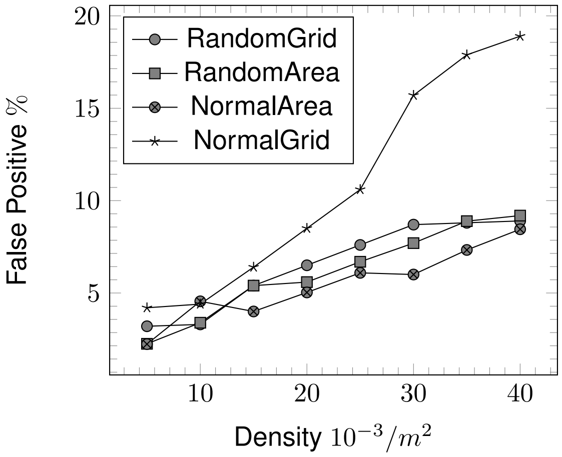

Figure 5. False positive vs. density ($Th = 10$).

not close enough according to so called close nodes. So we cannot see the false positive growing with increasing density. In NormalArea, the nodes are deployed with the normal distribution in the area. When the density increases, most the nodes originally close are still close. Therefore, the false positives do not grow with increasing density. From the above analysis, to minimize the false positives, a good distribution and a good threshold Th must be selected. To keep the false positives below 10%, the ideal distribution can be RandomGrid, RandomArea, and NormalArea with a maximum threshold Th value of 12.

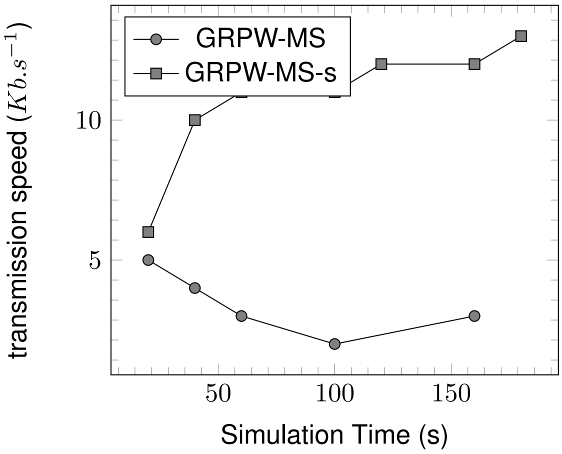

Figure 6. Wormhole attack analysis

In the network there are a set of attacking nodes which represents 20% of the network nodes , which are A_1 and A_z in the figure. For exemple , A_1 and A_2, which are the tunnel's two ends, will execute the wormhole attack. A_1 will tunnel all i t's hearing HELLO packets to A_2 , A_2 will also tunnel all the hearing HELLO packets to A_1 , then both of them will replay the HELLO packets. We simulate the originate GRPW-MS protocol and the revised protocol GRPW-MS-S under the same condition. The results are shown in Fig. 6. From the figure, we can see that the lower line is a zero line which simulate the originate GRPW-MS protocol. The zero means that No can not find a right route to send the packet , so Nu receives nothing . All these happened cases are caused by wormhole attackers making misbelieving being its neighbor. The upper line is the result of simulating the revised GRPW-MS-S protocol, we can found at first node also can not find the right route , but after evaluating some neighbor's trustiness, eache node start to choose another route to send the packet, after many times trying and evaluating, No finally find a stable route to Nal, so in the figure it shows that the transmitting rate is going to keep stable with time, and after $20s$, it keeps about $10.0kb/s$.

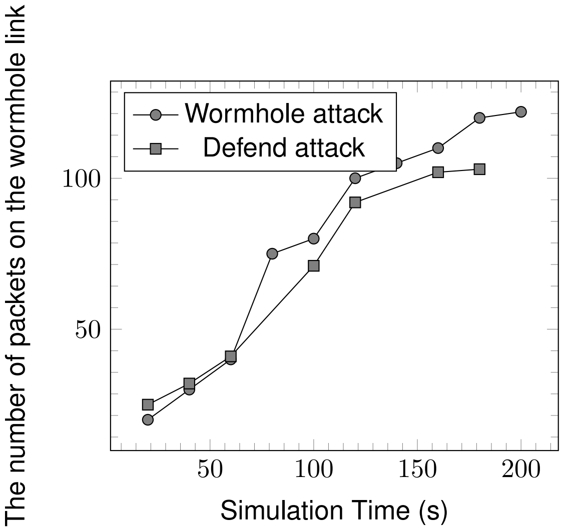

Figure 7. The number of packets on the wormhole link against time for GPRW-MS-S

Finally, we evaluate the defending effectiveness after detecting the wormhole attack. When the wormhole attack is initiated, the surrounding packets would transfer from the original route to this highquality wormhole link. As shown in the Fig.7, the dot curve indicates that the number of packets on the wormhole link dramatically increases after the wormhole attack; when the defending nodes begin to take defensive measures, the square curve reveals that the number of packets on the wormhole link grows exponentially. Gradually, the wormhole link becomes congested and the metric of link decreases, which indicates that our algorithm's defense against wormhole is effective. Therefore, when the nodes conduct the neighbor discovery, they will remove the malicious nodes from their respective neighbor lists and the wormhole link gets eliminated.

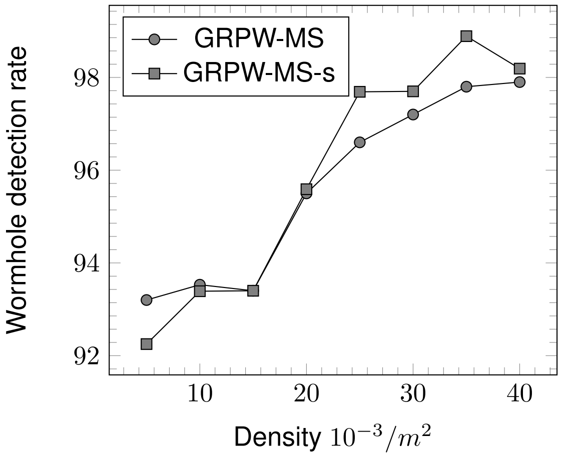

Figure 8. Wormhole detection rate against density ($Th = 10$).

we evaluate the algorithm's performance on detecting wormhole by varying the length of wormhole link. Fig.8 reveals the relationship between wormhole detection rate and the density. In

Fig.8, we can find that two algorithms both have a high detection rate. In GRPW-MS algorithm, when density varies from 5 to 20, the detection rate has a slight downward trend. When density continues to increase, the detection rate levels off, maintaining at about 0.955. By contrast, in our proposal, the detection rate shows an upward trend with the increase of density. Moreover, when density is greater than 10, the detection rate of our algorithm GRPW-MS-S is higher than that of GRPW-MS. The reason is that the longer the wormhole link is, the more hops the packets have to pass from source to destination if packets are transmitted through the normal link. But if there exists a wormhole link, the hops between source and destination would dramatically decrease and thus make the wormhole attack effect much more significant. So, according to our algorithm, we can easily detect the wormhole attack and thus get a high detection rate.

8. Conclusions

Because of the wireless medium's openness, every node can hear the neighbor's radio without being detected. When two or more malicious nodes construct one or more wormholes, they can destroy the entire Network by disrupting the routing protocol, especially to GRPW-Mus protocols. In this paper we introduced a trust model to evaluate the trustiness of ”a node is the neighbor” in GRPW-Mus protocol. From the trustiness calculating, the node can get the right route instead of choosing the route caused by wormhole attack. This scheme can run with no need for network synchronization and GPS devices. But the scheme is based on trust evaluation, which predicts the future events by collecting the past events, so the trust evaluated by the node lags behind the attacks. In future work, we will work on how to secure the trustiness message transmission and how to get the recommended path in trust graph. We also take the node's mobility into consideration, because when the network topology changing fast, the route will change fast, which means the trust model should keep track with it.

References

[1] S. K. Jangir and N. Hemrajani, “Evaluation of black hole, wormhole and sybil attacks in mobile ad-hoc networks,” in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, ser. ICTCS ’16. New York, NY, USA: ACM, 2016, pp. 74:1–74:6. [Online]. Available: http://doi.acm.org/10.1145/2905055.2905133

[2] R. Mudgal and R. Gupta, “An efficient approach for wormhole detection in manet,” in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, ser. ICTCS ’16. New York, NY, USA: ACM, 2016, pp. 29:1–29:6. [Online]. Available: http://doi.acm.org/10.1145/2905055.2905235

[3] M. Cinque, A. Coronato, A. Testa, and C. Di Martino, “A survey on resiliency assessment techniques for wireless sensor networks,” in *Proceedings of the 11th ACM International Symposium on Mobility Management and Wireless Access*, ser. MobiWac ’13. New York, NY, USA: ACM, 2013, pp. 73–80. [Online]. Available: http://doi.acm.org/10.1145/2508222.2508235

[4] I. Ouafaa, L. Jalal, K. Salah-ddine, and E. H. Said, “The comparison study of hierarchical routing protocols for ad-hoc and wireless sensor networks: A literature survey,” in *Proceedings of the The International Conference on Engineering & MIS 2015*, ser. ICEMIS ’15. New York, NY, USA: ACM, 2015, pp. 32:1–32:8. [Online]. Available: http://doi.acm.org/10.1145/2832987.2833039

[5] K. S. Hamza and F. Amir, “Centralized clustering evolutionary algorithms for wireless sensor networks,” in *Proceedings of the 10th International Conference on Informatics and Systems*, ser. INFOS ’16. New York, NY, USA: ACM, 2016, pp. 273–277. [Online]. Available: http://doi.acm.org/10.1145/2908446.2908493

[6] G. M. Dias, B. Bellalta, and S. Oechsner, “A survey about prediction-based data reduction in wireless sensor networks,” *ACM Comput. Surv.*, vol. 49, no. 3, pp. 58:1–58:35, Nov. 2016. [Online]. Available: http://doi.acm.org/10.1145/2996356

[7] R. A. Uthra and S. V. K. Raja, “Qos routing in wireless sensor networks&mdash;a survey,” *ACM Comput. Surv.*, vol. 45, no. 1, pp. 9:1–9:12, Dec. 2012. [Online]. Available: http://doi.acm.org/10.1145/2379776.2379785

[8] M. Hammoudeh, “Applying wireless sensor networks to solve real-world problems,” in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, ser. IPAC ’15. New York, NY, USA: ACM, 2015, pp. 1:1–1:1. [Online]. Available: http://doi.acm.org/10.1145/2816839.2816935

[9] M. P. Sousa, A. Kumar, M. S. Alencar, and W. T. Lopes, “Performance evaluation of a selective cooperative scheme for wireless sensor networks,” in *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, ser. PE-WASUN ’09. New York, NY, USA: ACM, 2009, pp. 85–92. [Online]. Available: http://doi.acm.org/10.1145/1641876.1641892

[10] F. Mohammed, E. F. Youssef, H. Abedehalim, and E. Abdellah, “Investigating the impact of black-hole attack on hierarchical protocols and direct transmission in wsn,” in *Proceedings of the International Conference on Internet of Things and Cloud Computing*, ser. ICC ’16. New York, NY, USA: ACM, 2016, pp. 78:1–78:4. [Online]. Available: http://doi.acm.org/10.1145/2896387.2900330

[11] Y. Sun, X. Wang, and X. Zhou, “Jamming attack in wsn: A spatial perspective,” in *Proceedings of the 13th International Conference on Ubiquitous Computing*, ser. UbiComp ’11. New York, NY, USA: ACM, 2011, pp. 563–564. [Online]. Available: http://doi.acm.org/10.1145/2030112.2030214

[12] K.-L. Tsai, M. Ye, and F.-Y. Leu, “Secure power management scheme for wsn,” in *Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats*, ser. MIST ’15. New York, NY, USA: ACM, 2015, pp. 63–66. [Online]. Available: http://doi.acm.org/10.1145/2808783.2808790

[13] M. Klonowski and M. Koza, “Countermeasures against sybil attacks in wsn based on proofs-of-work,” in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec ’13. New York, NY, USA: ACM, 2013, pp. 179–184. [Online]. Available: http://doi.acm.org/10.1145/2462096.2462125

[14] Y. Sabri and N. E. Kamoun, “Geographic routing in wireless sensor networks based on a partitioned architecture,” *International Journal of Computer Applications*, vol. 153, no. 5, pp. 1–8, Nov 2016. [Online]. Available: http://www.ijcaonline.org/archives/volume153/number5/26396-2016912059

[15] A. Laouid, M.-L. Messai, A. Bounceur, R. Euler, A. Dahmani, and A. Tari, “A dynamic and distributed key management scheme for wireless sensor networks,” in *Proceedings of the International Conference on Internet of Things and Cloud Computing*, ser. ICC ’16. New York, NY, USA: ACM, 2016, pp. 70:1–70:6. [Online]. Available: http://doi.acm.org/10.1145/2896387.2900322

[16] Y. Xu, Y. Ouyang, Z. Le, J. Ford, and F. Makedon, “Analysis of range-free anchor-free localization in a wsn under wormhole attack,” in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, ser. MSWiM ’07. New York, NY, USA: ACM, 2007, pp. 344–351. [Online]. Available: http://doi.acm.org/10.1145/1298126.1298185

[17] L. Yao, L. Kang, P. Shang, and G. Wu, “Protecting the sink location privacy in wireless sensor networks,” *Personal Ubiquitous Comput.*, vol. 17, no. 5, pp. 883–893, Jun. 2013. [Online]. Available: http://dx.doi.org/10.1007/s00779-012-0539-9

[18] T. Dimitriou and A. Giannetsos, “Wormholes no more? localized wormhole detection and prevention in wireless networks,” in *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 334–347. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13651-1_24

[19] H. Sarbazi-Azad and M. Ould-Khaoua, “A simple mathematical model of adaptive routing in wormhole k-ary n-cubes,” in *Proceedings of the 2002 ACM Symposium on Applied Computing*, ser. SAC '02. New York, NY, USA: ACM, 2002, pp. 835–839. [Online]. Available: http://doi.acm.org/10.1145/508791.508954

[20] W. Wang and B. Bhargava, “Visualization of wormholes in sensor networks,” in *Proceedings of the 3rd ACM Workshop on Wireless Security*, ser. WiSe '04. New York, NY, USA: ACM, 2004, pp. 51–60. [Online]. Available: http://doi.acm.org/10.1145/1023646.1023657

[21] T. Arici, T. Akgun, and Y. Altunbasak, “A prediction error-based hypothesis testing method for sensor data acquisition,” *ACM Trans. Sen. Netw.*, vol. 2, no. 4, pp. 529–556, Nov. 2006. [Online]. Available: http://doi.acm.org/10.1145/1218556.1218560

[22] T. Dimitriou and A. Giannetsos, “Wormholes no more? localized wormhole detection and prevention in wireless networks,” in *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 334–347. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13651-1_24

[23] D. Dong, M. Li, Y. Liu, X.-Y. Li, and X. Liao, “Topological detection on wormholes in wireless ad hoc and sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1787–1796, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1109/TNET.2011.2163730

[24] T. Minohara and K. Nishiyama, “Poster: Detection of wormhole attack on wireless sensor networks in duty-cycling operation,” in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '16. USA: Junction Publishing, 2016, pp. 281–282. [Online]. Available: http://dl.acm.org/citation.cfm?id=2893711.2893773

[25] K.-F. Krentz and G. Wunder, “6lowpan security: Avoiding hidden wormholes using channel reciprocity,” in *Proceedings of the 4th International Workshop on Trustworthy Embedded Devices*, ser. TrustED '14. New York, NY, USA: ACM, 2014, pp. 13–22. [Online]. Available: http://doi.acm.org/10.1145/2666141.2666143

[26] S. P. Marsh, “Formalising trust as a computational concept,” Tech. Rep., 1994.

[27] A. Varga and R. Hornig, “An overview of the omnet++ simulation environment,” in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 60:1–60:10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1416222.1416290

[28] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, “Simulating wireless and mobile networks in omnet++ the mixim vision,” in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 71:1–71:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=1416222.1416302