Detecting Selective Forwarding Attacks in Wireless Sensor Networks *

Bo Yu[1,2] Bin Xiao[1]

[1]Hong Kong Polytechnic University
Dept. of Computing
Hung Hom, Kowloon, Hong Kong
{csbyu, csbxiao}@comp.polyu.edu.hk

[2]Fudan University
Dept. of Computer Science and Engineering
Shanghai, 200433, P.R.China
boyu@fudan.edu.cn

Abstract

Selective forwarding attacks may corrupt some mission-critical applications such as military surveillance and forest fire monitoring. In these attacks, malicious nodes behave like normal nodes in most time but selectively drop sensitive packets, such as a packet reporting the movement of the opposing forces. Such selective dropping is hard to detect. In this paper, we propose a lightweight security scheme for detecting selective forwarding attacks. The detection scheme uses a multi-hop acknowledgement technique to launch alarms by obtaining responses from intermediate nodes. This scheme is efficient and reliable in the sense that an intermediate node will report any abnormal packet loss and suspect nodes to both the base station and the source node. To the best of our knowledge, this is the first paper that presents a detailed scheme for detecting selective forwarding attacks in the environment of sensor networks. The simulation results show that even when the channel error rate is 15%, simulating very harsh radio conditions, the detection accuracy of the proposed scheme is over 95%.

1. Introduction

Wireless Sensor Networks (WSNs) are ideal candidates for monitoring environments in a wide variety of applications such as military surveillance and forest fire monitoring [7]. In such a network, a large number of sensor nodes are deployed over a vast terrain to detect events of interest (e.g., enemy vehicles, outbreaks of forest fires), and to deliver data reports to the base station over multi-hop wireless paths. The node-patterned deployment of WSNs, however, can be the focus of certain types of malicious attack. One such strategy is the selective forwarding attack, first proposed by Karlof [2]. In such attacks, a malicious node selectively drops sensitive packets, for example, a packet reporting the enemy tank movements. Selective forwarding attacks are typically most effective when the attacking nodes are explicitly included on the path of a data flow. They can corrupt a number of existing routing protocols such as TinyOS beaconing, Directed Diffusion [9], GPSR [10], GEAR, and clustered based protocols, especially when they are used in combination with other attacks such as wormhole and sinkhole attacks.

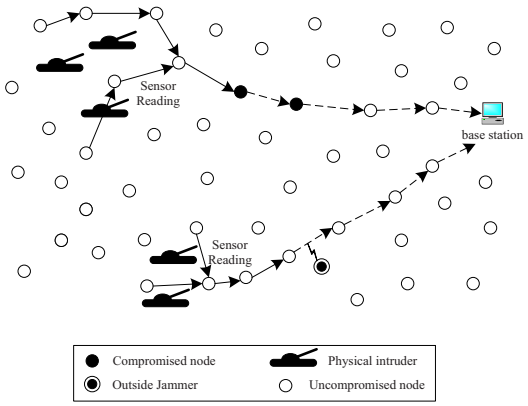
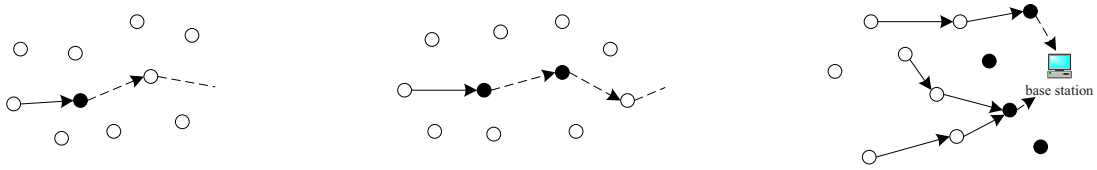


Figure 1. An example sensor network under selective forwarding attacks.

Karlof *et al.*[2] suggested countering selective forwarding by using multipath forwarding. However, multipath forwarding also suffers from several drawbacks. First, communication overheads increase dramatically as the number of paths increases. Second, multiple paths ultimately join up in the area neighboring the base station, so if nodes around the base stations are compromised, selective forwarding is still applicable. Finally, the multipath forwarding shows poor security resilience. To compromise the system, an ad-

*This work is partially supported by HK POLYU A-PA2F and ICRG A-PG52.

1-4244-0054-6/06/$20.00 ©2006 IEEE



(a) Single malicious node (b) Two consecutive malicious nodes (c) Surrounding malicious nodes

Figure 2. Deployment of malicious nodes.

versary only needs to ensure the presence of one compromised node in each path. Traditional transport layer protocols [11,12] for WSNs also fail to guarantee that packets are not maliciously dropped. They are not designed to deal with malicious attacks.

In this paper, we propose a lightweight security scheme that detects selective forwarding attacks by using a multi-hop acknowledgement technique that increases detection accuracy yet lowers overhead. The scheme allows both the base station and source nodes to collect attack alarm information from intermediate nodes. This means that even when the base station is deafened by surrounding malicious nodes, the source nodes can still make decisions and responses. The scheme can efficiently obtain those alarm information whenever intermediate nodes in a packet forwarding path detect any malicious packet dropping. Simulation results show that the communication overhead of our scheme is usually less than 2 times the overhead of the common one-path packet delivery process, and the detection accuracy is over 95% even when the channel error rate is a harsh 15%. To the best of our knowledge, this is the first paper that presents a detailed detection scheme in response to selective forwarding attacks.

The remainder of this paper is organized as follows. Section 2 introduces the attack model and our design goals. Section 3 presents our detection scheme based on multi-hop acknowledgement in detail. Section 4 first proposes several evaluation metrics for our detection scheme and then shows the simulation results in terms of these metrics. Finally, we introduce the related work in Section 5, and conclude our work in Section 6.

2. Attack Model

We consider a military application of sensor networks for reconnaissance of opposing forces, as shown in Figure 1. Each sensor node is battery-powered and has limited sensing, computation and wireless communication capabilities. When the activities of the opposing forces such as tank movement are detected, sensor readings are aggregated to generate a report, which will be forwarded to the base station through multi-hops. The sink is a data collection center equipped with sufficient computation and storage capabilities. Once the base station receives the report, it can take action by, for example, sending soldiers or missiles to the target field.

In a military application such as this, prompt detection and reporting of each relevant event in the field are important, but these processes can be easily corrupted by *selective forwarding attacks*. In such attacks, malicious nodes may refuse to forward certain packets and simply drop them, ensuring that they are not propagated any further. An adversary will not, however, drop every packet. To avoid raising suspicions, the adversary instead selectively drops packets originating from a few selected nodes and forwards the remaining traffic. As shown in Figure 1, the adversary may attack in two ways, from inside the network via compromised nodes or from outside the network by jamming the communication channels between uncompromised nodes.

Figure 2 shows three basic ways in which malicious nodes can be distributed in a network for different tactical purposes. Figure 2(a) shows a single malicious node located in the middle of a forwarding path. This node can selectively forward packets to the base station. Figure 2(b) shows two or more malicious nodes chained along a forwarding path. This can make it more difficult to detect packet dropping. Figure 2(c) shows a number of compromised nodes surrounding a base station. This arrangement can be used to deafen a base station by refusing to forward any packets at all.

In this paper, our goal is to design a scheme that detects selective forwarding attacks and identifies malicious nodes. Once the ids of suspect nodes are known, routing protocols can exclude them from routing paths. Furthermore, professionals can be sent to the battlefield to examine the nodes physically and even physically remove the compromised nodes. The detection scheme should have the following properties. First, the scheme should be able to quickly detect any malicious packet dropping. Second, detection accuracy should be guaranteed even when radio conditions are poor. Finally, the scheme should cause little additional communication overhead.

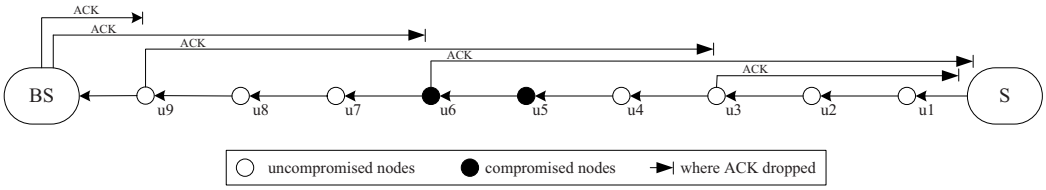


Figure 3. An example of multi-hop acknowledgement with $ACK\_Span = 3$,$ACK\_TTL = 6$. Node $u3$,$u6$,$u9$ are ACK nodes, which are required to send out ACK packets.

3. A Multi-hop Acknowledgement-Based Detection Scheme

In this section, we present the design of our multi-hop acknowledgement-based detection scheme. In our scheme, each intermediate node along the forwarding path is in charge of detecting malicious nodes. If an intermediate node detects the misbehavior of its downstream (upstream) nodes, it will generate an alarm packet and deliver it to the source node (the base station) through multiple hops. In this paper, downstream denotes the direction toward the base station, and upstream denotes the direction toward the source node. The base station and the source node can then use more complicated IDS (Intrusion Detection System) algorithms to make decisions and responses.

3.1. Assumptions

The following five assumptions are appropriate to use of the proposed detection scheme in a mission-critical application such as military reconnaissance, as opposed to a civilian application such as temperature monitoring. Our first assumption is that during the deployment phase each sensor can acquire its geographical position and loosely synchronize its time with the base station. Secure positioning and time synchronization discussed in [15,16,17] are also required by other purposes in a military mission, so our assumption does not add additional overhead. Note that our scheme can function over various existing routing algorithms but does not rely on geographic routing. Second, we assume that the adversary cannot successfully compromise a node during the short deployment phase. Some existing work [7] has made similar assumptions and argued that such attacks can indeed be prevented in real-life scenarios when appropriate network planning and deployment keep away attackers during the bootstrapping process. Third, we assume that malicious nodes, in order to allay suspicions, selectively drop only a small proportion of all packets passing by rather than every packet. Fourth, we assume that each node shares a master secret key with the base station and a node can establish a pairwise key with another node that is multiple hops away. Finally, we assume that routing and transport protocols such as Directed Diffusion[9] and PSFQ[10] have been implemented in sensor nodes. Our scheme can function over these protocols.

Although the routing layer of WSNs is threatened by various attacks, here we are considering only selective forwarding attacks. It is out of the scope of the paper to consider the issue of verifying whether an sensor report is modified or injected. The interested will find discussions of this in [4,5,6,7]. Nor do we consider the detection of link-layer jamming attacks [14], which are also able to cause packet loss.

3.2. Node Initialization and Deployment

Before deployment, the key server loads every node with a unique secret key and a symmetric bivariate polynomial $f(u, v)$.

The unique key is shared with each sensor node and the base station and can be used to encrypt sensor reports and generate MACs (Message Authentication Codes) for the reports.

The symmetric bivariate polynomial is used to establish a pairwise key between any two sensor nodes in the network. Before deployment, the key server generates a symmetric bivariate k-degree polynomial $f(u, v) = \Sigma_{i,j=0}^{k} a_{ij} u^i v^j$ over a finite field F_q, where q is a prime number that is large enough to accommodate a cryptographic key. A polynomial $f(u, v)$ is said to be symmetrical if $f(u, v) = f(v, u)$. The key server then loads each node with this polynomial. After deployment, each node regenerates a polynomial $g(x)$ using its node id: $g(x) = f(id, x)$, and erases $f(u, v)$ from its memory forever. Suppose that node a owns $g_a(x) = f(a, x)$, node b owns $g_b(x) = f(b, x)$, and node a wants to send a packet to node b. First node a calculates $k_1 = g_a(b)$ as the key to generate a MAC for the packet and then sends the packet together with the MAC and its id to node b. After receiving the packet, node b calculates $k_2 = g_b(a)$, ($k_1 = k_2$), as the decryption key and verify the MAC. If the verification is passed, node b believes the packet comes from the authentic node a.

During deployment, each nodes tries to find its downstream and upstream nodes which might be multi-hop away. The node just need to save its neighbors' ids in memory for generating a pairwise key in the future. The id information can be piggybacked on existing routing messages such as TinyOS beaconing or interest and reinforcement messages in Directed Diffusion [9]. As a result, little memory and communication overhead will be incurred for these operations.

Please note that we assume it is secure in the deployment phase. After the original bivariate polynomial $f(u, v)$ is erased from the node's memory, it will be difficult for the adversaries to regenerate $f(u, v)$. When used in key establishment, symmetric bivariate polynomials have been proved to be unconditionally secure as long as no more than k nodes are compromised [4].

3.3. OHC-based One-to-Many Authentication

In this subsection, we use OHC(Oneway Hash Chain) to establish one-to-many authentication among sensor nodes, which might be multiple hops away. Compared with pairwise key-based authentication, the OHC-based one-to-many authentication allows the reduction of both communication and computation overhead.

OHC is a sequence of numbers, generated by a publicly known one-way function F. To generate the one-way key chain, randomly choose the last key K_n of the chain, and repeatedly apply F to compute all other keys: $K_i = F(K_{i+1})$. OHC was proposed in [19] and has since been widely used on account for its simplicity and efficiency in one-to-many authentication, for example, in [6].

In our scheme, each node generates and maintains a one-way hash chain $< K_0, K_1, ..., K_n >$, where $Ki = F(K_{i+1})$. Each node can send K_0 to its upstream or downstream nodes during the neighbor discovery phase. The pairwise key generated by bivariate polynomial will encrypt the K_0 message. When node u wants to send a message to node v, it just sends out: $\{data, K_i, MAC_{Ki}(data)\}$, where $i \geq 1$ and i increases by one after one message is sent out. To authenticate the message, node v just needs to verify whether $K_{i'} = F(F..F(K_i))$, where $K_{i'}$ is the previous OHC number node v receives, and the number of iterative calls of $F()$ should be limited. If the verification fails, node v may request that node u updates his OHC numbers. For reasons of space, we omit a detailed discussion of the generation and maintenance of OHC.

During deployment, each node also tries to exchange its first OHC number with its upstream and downstream nodes. This process can be combined with the id exchange operations introduced in Section 3.2.

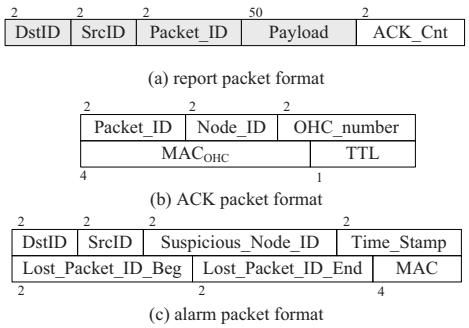


Figure 4. Packet formats.

3.4. Upstream Detection Process

In this subsection, we describe how an intermediate node detects suspect nodes. Our detecting task consists of *upstream detection* and *downstream detection*. The uncompromised nodes upstream of malicious nodes take charge of the upstream detection task, and the uncompromised nodes downstream of malicious nodes take charge of the downstream detection task.

Three packets, *report packet*, *ACK packet*, and *alarm packet*, are used in our scheme. The suggested fields and corresponding byte lengths for each packet are presented in Figure 4.

First we consider the upstream detecting task. A source node generates a report packet, when it detects a special event, e.g. tank movement noise. The report will be forwarded toward the base station hop-by-hop. The initial ACK_Cnt is set to ACK_Span, which is a predefined metric. When each intermediate node receives the report packet, it first saves the report packet in its cache, decreases the ACK_Cnt by one, or resets ACK_Cnt to its initial value ACK_Span if ACK_Cnt equals to 0 already, and then forwards the report packet to the next downstream node. Meanwhile, if the node finds ACK_Cnt is equal to 0, it generates an ACK packet, where the TTL in the ACK packet is initially set to ACK_TTL, which is also a predefined metric. The node sends the ACK packet to the upstream node where the previous report packet comes from. The ACK packet will traverse multiple hops until TTL is decreased to 0, following the same path as traversed by the previous report but in the opposite direction. We call the nodes which are required to send out ACK packets *ACK nodes*, such as nodes $u3$,$u6$,and $u9$ in Figure 3.

After an intermediate node forwards a report packet to the downstream neighbor, it waits for ACK packets which will be returned by the downstream neighbor. If less than t ACK packets are returned within time T_{ack}, the node suspects that the previous report packet might have been dropped by a malicious node downstream. The intermediate

node then generates an alarm packet. $DstID$ in the alarm packet is the source node id. Both $Lost\_Packet\_ID\_Beg$ and $Lost\_Packet\_ID\_End$ are the lost report packet id. $Time\_Stamp$ is set to the current system time in the sensor node. The node chooses the next downstream node to be the suspect node and set it in the $Suspicious\_Node\_ID$ field in the alarm packet. The alarm packet then is forwarded through multiple hops to the source node. There might be more than one alarm packet generated during the process of delivering one report packet. Some of these alarm packets will be false alarms, because the ACK packets might have been dropped due to harsh radio conditions. However, when the source node finally receives all the alarm packets, it can easily remove the false alarms.

The parameters, ACK_Span and ACK_TTL, provide a tradeoff between detection capability and communication overhead. Bigger ACK_Span and ACK_TTL help to increase the security resilient when a number of nodes in a path is compromised, but also increase the communication overhead and the waiting time before enough ACK packets arrive. We evaluate these parameters in a simulation in Section 4.

It is possible for a malicious node to fabricate an alarm packet but this would have only a limited effect because the malicious node can only set the next downstream node as the suspicious node. The source node will regard both the suspicious node specified in the alarm packet and the node which generates the alarm packet as malicious nodes (or threatened nodes). Note that it is unnecessary to distinguish between malicious nodes and threatened nodes near the malicious nodes as both malicious and threatened nodes should be excluded from the forwarding path once routing responses are made. Some existing works such as [3] propose similar precautions in response to malicious and threatened nodes.

Figure 3 provides an example of the operation of our upstream detection mechanism. Suppose that node $u5$ and $u6$ are compromised nodes, and node $u5$ drops a report packet coming from the source node. First we consider node $u2$. After forwarding the report packet to node $u3$, node $u2$ waits for ACK packets. Suppose that node $u2$ receives an ACK packet from node $u3$ but does not receives an ACK from $u6$ within time T_{ack}. Node $u2$ will set node $u3$ as the suspect node in its alarm packet. The alarm packet will be forwarded through multiple hops to the source node. Next we consider node $u4$. Node $u4$ receives no ACK packets. It sets the next downstream node, node $u5$, as the suspect node. Node $u4$ also generates an alarm packet and sends it to the source node. Finally, we consider node $u6$, which fabricates an alarm packet which says node $u7$ is the suspect node. The alarm packet is also delivered to the source node. In this way, the source node will receive 3 alarm packets for the same report packet. However, because it can be sure that the report packet did arrive at node u4, it's easy for the source node to remove the false alarms from nodes $u2$ and $u4$. The source node finally concludes nodes $u6$ and $u7$ as the malicious nodes or threatened nodes.

3.5. Downstream Detection Process

We next consider the downstream detection task. If an intermediate node receives a report packet which has a discontinuous $Packet\_ID$ for a specific source node, packet loss might have occurred. The node generates an alarm packet, in which, $Lost\_Packet\_ID\_Beg$ and $Lost\_Packet\_ID\_End$ describe the range of the lost $Packet_ID$s, and $Suspicious\_Node\_ID$ is set to the upstream node where the report with the discontinuous $Packet\_ID$ came from. The alarm packet will be forwarded through multiple hops to the base station. The discontinuity of $Packet_ID$s might be caused by a malicious upstream node, a nearby outside jammer, or even by routing topology changes. Thus it is likely that the alarm packet is a false alarm. However, when the base station ultimately receive all the report packets, it is easy for the base station to remove false alarms.

3.6. Several Other Issues

ACK nodes are critical points along the forwarding path, such as node $u3$, $u6$, and $u9$ in Figure 3. If these nodes are compromised, the adversaries can fabricate ACK packets. It is thus important that the intermediate nodes which act as ACK nodes should not be fixed. Upstream nodes should require that ACK packets for the current report come from different ACK nodes than for the last report. In order to achieve this goal, we let the source node set the ACK_Cnt field in each report in manner of a descending (or ascending) counter. The initial ACK_Cnt value at the source node will be cycled between 0 and $(ACK\_Span - 1)$. In this way, each node has an equal chance of becoming an ACK node as well as an equal chance of being compromised.

When global packet loss information from various sources and paths are collected at the base station and the source node, more complicated IDS(Intrusion Detection System) algorithms such as game theory or statistical analysis can be implemented at the base station or even at the source node. After monitoring the packet loss information for a period of time, the base station and the source node may make decisions and take actions allowing the routing layer exclude the malicious nodes from the forwarding path.

WSN resilience can be improved by integrating redundant and intrusion detection approaches. Its quite difficult to prevent a compromised yet undetected intermediate node from dropping a packet going through it. One answer might be to allow the source node to decide how a packet is deliv-

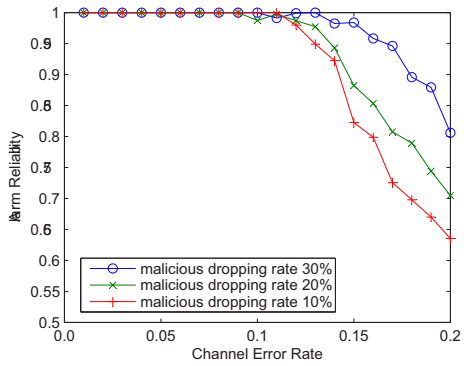


Figure 5. Alarm reliability when the channel error rate increases from 1% to 20%.

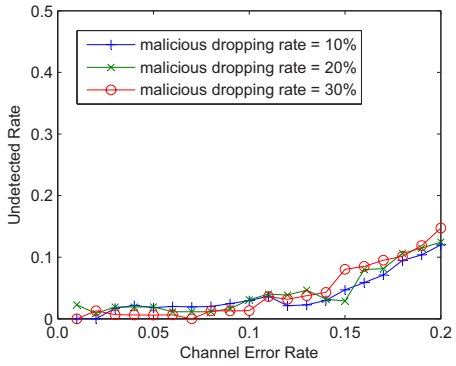


Figure 6. Undetected rate when the channel error rate increases from 1% to 20%.

ered. If the packet is a very important one, the source node can have the packet delivered via multipath forwarding, or even flooding. If the packet is just a routine message, the packet can be delivered via one-path forwarding with intrusion detection.

4. Performance Evaluation

In this section, we evaluate the performance, such as the detection accuracy and communication overhead of our scheme through simulations. We use a field size of $2000 \times 2000m^2$ where 400 nodes are uniformly distributed. One stationary sink and one stationary source sit on opposite sides of the field, with about 20 hops in between. We carry out a simulation event in which the source generates 500 reports in total and one report is sent out every two seconds. Packets can be delivered hop-by-hop at 19.2 Kbps. In order to avoid detection, the malicious nodes drop only part of the packets passing by. To make our scheme more resilient in poor radio conditions, we implement a hop-by-hop transport layer retransmission mechanism beneath our scheme, which is quite similar to that in PSFQ[11]. The retransmission limit is 5 by default. The channel error rate is 10% by default , which is usually regarded as a rather harsh radio condition. Each simulation runs 10 times and the result shown is an average of these runs. We first define 3 metrics and then provide our simulation results for these metrics.

4.1. Evaluation Metrics

The first two proposed metrics evaluate the detection accuracy of our scheme. The third evaluates the communication overhead.

Alarm reliability measures the ratio of the number of detected maliciously-dropped packets to the total number of lost packets detected including those lost due to poor radio conditions.

Undetected rate measures the ratio of the number of undetected maliciously-dropped packets to the total number of maliciously-dropped packets.

Relative communication overhead measures the ratio of the total communication overhead in a system that incorporates our detection scheme against a system that does not.

4.2. Detection Accuracy

In this section, we study how the detection accuracy of our scheme is affected by the channel error rate, the packet retransmission mechanism and the number of compromised nodes. Our first simulation shows the impact of the channel error rate on alarm reliability. Figure 5 illustrates that the channel error rate could be the main cause affecting alarm reliability. If the condition of communication links is presumed to be perfect, packet loss must be the result of malicious dropping. Thus channel error rate is a good indicator of alarm reliability. Indeed, as shown in Figure 5, when the channel error rate is less than 10%, alarm reliability is close to 100%. Alarm reliability falls rapidly as the channel error rate increases over 10% because it is difficult to distinguish packet loss due to malicious dropping from that due to poor radio conditions. Interestingly, Figure 5 also suggests that a larger malicious dropping rate will increase the alarm reliability, on the intuition that malicious nodes dropping more packets are easier to detect. On the whole, our detection scheme works well to achieve over 80% reliability of alarms, even when the channel error rate is 15%, which is usually regarded as rather harsh radio conditions.

Our second simulation tests the impact of the channel error rate and compromised nodes on the system undetected

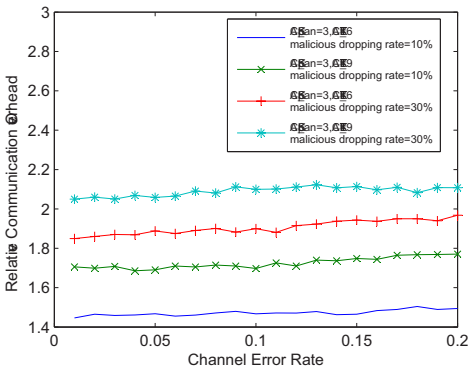


Figure 7. Relative Communication Overhead when the channel error rate increases from 1% to 20%

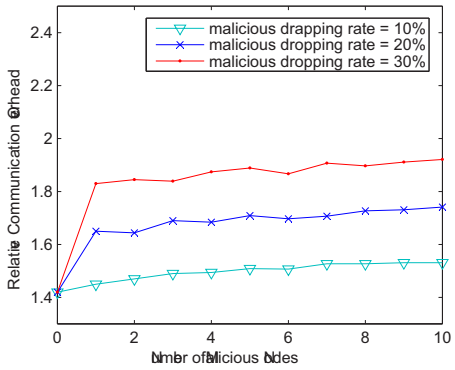


Figure 8. Relative Communication Overhead in terms of number of compromised nodes.

rate. The undetected rate is mainly affected by two factors, the channel error rate and how many nodes are compromised. A high channel error rate will cause alarm packets to be dropped even after several retransmissions. Similarly, if there are multiple compromised nodes along the forwarding path, some compromised nodes might drop alarm packets to prevent other compromised nodes from being detected. Note that if malicious packet dropping is detected by a node in the middle of a path, but the alarm packets are ultimately not delivered to the base station or the source node, we still regard this as a case of undetection. Figure 6 presents the results for the undetected rate as we increase the channel error rate. The undetected rate increases as the channel error rate increases. This is probably the result of alarm packets being dropped as retransmission times reach their limit under harsh radio conditions. The results also show that various malicious dropping rates do not greatly impact the undetected rate.

4.3. Communication Overhead

We compare the communication overhead of two systems, one incorporating the proposed selective forwarding attacks detection scheme and one that does not. Their ratio is denoted as the relative communication overhead. This ratio can help to us compare our scheme, in which data delivery is based on one-path, with other anti-selective-forwarding approaches such as the multipath approach mentioned in [2].

Figure 7 provides a more detailed view of the aggregated effects of a malicious dropping rate, ACK_Span, and ACK_TTL on relative communication overhead. As the four curves show, when the channel error rate increases, relative communication overhead increases very little. An increased malicious dropping rate leads to increased relative communication overhead, as more alarm packets are generated and forwarded. A bigger ACK_TTL value can cause relative communication overhead to increase because an ACK packet traverses more hops before it is dropped.

The number of malicious nodes also impact the relative communication overhead. In our simulation, there are about 20 hops in the forwarding path between the base station and the source node. We suppose that the number of malicious nodes in the forwarding path increases from 0 to 10. Figure 8 shows that the relative communication overhead increases smoothly as the number of malicious nodes increases. Please note that when the network contains no malicious nodes, the communication overhead is only about 1.4 times of the basic one-path delivery approach, which is apparently less than any existing multipath approach such as [2,3].

5. Related Work

WSN security has been studied in recent year in a number of proposals. Zhang and Lee [1] are among the first to study the problem of intrusion detection in wireless ad hoc networks. Karlof *et al.* [2] analyzes attacks against sensor network routing protocols, points out possible ways of defense and the author suggests a possible way to counter selective forwarding attacks by using multipath routing. Deng *et al.* [3] proposes INSENS, an intrusion-tolerant scheme based on multipath routing. These schemes [2,3] are all based on redundant routing.

En-route filtering of injected false data in sensor networks has been studied recently [4,5,6,7]. Zhu *et al.* [4] proposes an interleaved key scheme, in which member nodes and intermediate nodes set up interleaved keys using randomly pre-distributed keys. The SEF scheme [5] proposed by Ye *et al.* tries to filter false data by a probabilistic approach. Random keys are shared between the intermedi-

ate nodes and the source nodes in a sensor node group or cluster. Intermediate nodes can verify the MACs generated by the source nodes before forwarding packets. Yang *et al.* [7] presents a more resilient approach based on location-binding keys. However, in his scheme, the relative position between source nodes and the base station is static. His scheme will be inefficient, if there are more than one base station, or the base station is mobile.

6. Conclusion

In this paper, we propose a simple and efficient security scheme for detecting selective forwarding attacks. Unlike common approaches in which detection is implemented in the base station or in a central controller, our scheme lets both the base station and the source nodes have the capability to detect selective forwarding attacks. Thus even when the base station is temporarily deafened by adversaries, attacks can still be detected.

Through simulations, we observe that its difficult to distinguish packet loss due to compromised nodes from that due to outside jammers without jamming detection technique. Both of them share similar symptoms. Currently our scheme can only discriminate abnormal packet loss from channel error packet loss at a high detection ratio. As long as the malicious nodes, including compromised nodes and outside jammers, cause more packet loss than a normal node does at a certain channel error rate, the attacks are always detectable. In the future work, we plan to integrate the jamming detection techniques with our scheme, thus the original causes of abnormal packet loss can be find out.

References

[1] Y. Zhang and W. Lee. Intrusion Detection in Wireless Ad-Hoc Networks. In Proc. ACM MobiCom, pages 275-283, 2000.

[2] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA 03), pages 113-127, May 2003.

[3] J. Deng, R. Han and S. Mishra. INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks. In the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003), 2003.

[4] S. Zhu, S. Setia, S. Jajodia and N. Peng. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In Proc. of IEEE Symposium on Security and Privacy, pages 259-271, 2004.

[5] F. Ye, H. Luo, S. Lu and L. Zhang. Statistical en-route filtering of injected false data in sensor networks. In Proc. of IEEE InfoCom, 2004.

[6] J. Deng, R. Han and S. Mishra. Defending against Path-based DoS Attacks in Wireless Sensor Networks. In Proc. the 3rd ACM on the Security of Ad Hoc and Sensor Networks (SASN 2005), pages 89-96, 2005.

[7] H. Yang, F. Ye, Y. Yuan, S. Lu and W. Arbaugh. Toward resilient security in wireless sensor networks. In Proc. of ACM MobiHoc, pages 34-45, 2005.

[8] J. Deng, R. Han and S.Mishra. Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. In Proc. of International Conference on Dependable Systems and Networks, page 637, 2004.

[9] C. Intanagonwiwat, R. Govindan and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proc. of ACM MobiCom, pages 56-67, 2000.

[10] B. Karp and H.T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In Proc. of ACM MobiCom, pages 243-254, 2000.

[11] C. Y. Wan, A. T. Campbell, L. Krishnamurthy. PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks. In Proc. of the first ACM International Workshop on Wireless Sensor Networks and Applications, pages 1-11, 2002.

[12] Y. Sankarasubramaniam, O. B. Akan and I. F. Akyildiz. ESRT: Event to Sink Reliable Transport in Wireless Sensor Networks. In Proc. of ACM MobiHoc, pages 177-188, 2003.

[13] W. Xu, W. Trappe, Y. Zhang and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In Proc. of ACM MobiHoc, pages 46-57, 2005.

[14] Y. W. Law, L. V. Hoesel, J. Doumen, P. Hartel and P. Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. In Proc. of the 3rd ACM on the Security of Ad Hoc and Sensor Networks (SASN 05), pages 76-88, 2005.

[15] S. Capkun, J. Hubaux. Secure Positioning of Wireless Devices with Application to Sensor Networks. In Proc. IEEE InfoCom, 2005.

[16] S. Ganeriwal, S. Capkun, C. Han, M. B. Srivastava. Secure Time Synchronization Service for Sensor Networks. In Proc. of the 4th ACM workshop on Wireless Security (WiSe 06), pages 97-106, 2006.

[17] M. Manzo, T. Roosta, S. Sastry. Time Synchronization Attacks in Sensor Networks. In Proc. of the 3rd ACM on the Security of Ad Hoc and Sensor Networks (SASN 05), pages 107-116, 2005.

[18] C. Ozturk, Y. Zhang, W. Trappe. Source-Location Privacy in Energy-Constrained Sensor Network Routing. In Proc. of SASN 04, pages 88-93, 2004.

[19] L. Lamport. Constructing digital signatures from one-way function. in technical report SRI-CSL-98, SRI International, Oct. 1979.

[20] J. M. McCune, E. Shi, A. Perrig, M. K. Reiter. Detection of Denial-of-Message Attacks on Sensor Network Broadcasts. In Proc. of IEEE Symposium on Security and Privacy, pages 64-78, 2005.