Defending Malicious Collision Attacks in Wireless Sensor Networks

Phillip Reindl and Kendall Nygard
Department of Computer Science
North Dakota State University
Fargo, North Dakota, 58102, USA
e-mail: {phillip.reindl, kendall.nygard}@ndsu.edu

Xiaojiang Du
Dept of Computer and Information Sciences
Temple University
Philadelphia, Pennsylvania, 19122, USA
e-mail: dux@temple.edu

***Abstract*—Security is an important issue for sensor networks deployed in hostile environments, such as military battlefields. The low cost requirement precludes the use of tamper resistant hardware on tiny sensor nodes. Hence, sensor nodes deployed in open areas can be compromised and used to carry out various attacks on the network. In this paper, we consider the collision attack that can be easily launched by a compromised (or hostile) node: a compromised node does not follow the medium access control protocol and cause collisions with neighbor transmissions by sending a short noise packet. This attack does not consume much energy of the attacker but can cause a lot of disruptions to the network operation. Due to the wireless broadcast nature, it is not trivial to identify the attacker. In this paper, we propose a distributed scheme that is based on low-cost hardware and can effectively identify the source of a collision attack. Our scheme is based on analyzing physical-layer Received Signal Strength Index (RSSI) readings. We show that correct identification of an adversarial node can be achieved with greater than 85% accuracy. We further present a technique that degrades gracefully as the background noise increases.**

Keyword -security; collision attacks; sensor networks

I. INTRODUCTION

Security is an important and challenging issue in wireless sensor networks. A widely used attack model assumes that a sensor node does not have tamper resistant hardware (due to cost reason) and may be compromised in the field. A compromised node may be used to carry out various malicious attacks on the network. Several attacks on sensor nodes/networks have been studied, such as selective forwarding attack, wormhole attack, sinkhole attack, and Sybil attack [1].

In this paper, we study the malicious collision attack that can be easily launched by a compromised (or hostile) sensor node. In a collision attack, an attacker node does not follow the medium access control protocol and cause collisions with neighbor node's transmissions by sending a short noise packet. This attack does not consume much energy of the attacker but can cause a lot of disruptions to the network operation. Due to the wireless broadcast nature, it is not trivial to identify the attacker. In this paper, we present a distributed scheme that is based on low-cost hardware and can effectively identify the source of a collision attack. Basically, our scheme identifies the attacker by analyzing the physical-layer Received Signal Strength Index (RSSI) readings at neighbor nodes. RSSI readings are inherently unreliable due to the variability of the wireless medium. We overcome this unreliability through distributed sampling and centralized analysis of the RSSI readings. It has been shown that for multiple transmissions from a single source, the ratio of RSSI readings from neighbor nodes remains constant [2]. We leverage this fact to create unique fingerprints for nodes in a sensor network. The fingerprints are used for identifying the source of a collision attack with high confidence.

Most past work considered a homogeneous sensor network, where all nodes have the same (or similar) capabilities. In this work, we adopt a Heterogeneous Sensor Network (HSN) model that consists of a small number of powerful High-end sensors (H-sensors), in addition to a large number of small Low-end sensors (L-sensors). H-sensors have better capabilities than L-sensors in terms of communication, computation, energy supply, storage space, and other aspects. In our research, we take advantage of the strong capabilities of H-sensors for designing efficient and effective security schemes.

The rest of the paper is organized as follows: We discuss the related work in Section II, and describe the wireless fingerprinting framework in Section III. In Section IV, we present several effective schemes for identifying the source of a collision attack, and we report the experimental results in Section V. We conclude this paper in Section VI.

II. RELATED WORK

Demirbas and Song [2] developed a scheme for detecting the Sybil attack [1] by using the RSSI values from at least two detecting nodes. They showed that while the RSSI values for a given node vary greatly between transmissions, the ratio of RSSI values seen by two nodes for a given source is consistent. However, the goal in [2] is simply to determine whether two transmissions were from the same source, [2] did not present any practical techniques for determining the source of malicious transmission collisions in sensor networks. Furthermore, [2] only considered homogeneous sensor networks. Our work addresses a more difficult issue of identifying the source of a malicious collision. Also, we considered a HSN and utilized more powerful H-sensors.

Law, et al., [3] considered an attack where an outsider deploying a jamming network in the same area as the target network. They presented schemes for efficient jamming, with near 100% message suppression, while giving the jamming nodes a lifetime similar to the target network. Suggestions for more robust MAC layer protocols are given in [3].

A number of literatures have discussed methods for wireless fingerprinting by analyzing characteristics of the radio signal. Some are discussed below:

- Frequency shift – Due to the cost of manufacturing, every radio transmits at a slightly different frequency [4], and this can be used to identify a radio device.
- Transients – During power up and power down, wireless radios emit a noise signal. The noise signals are referred to as transients and are unique to each physical device [5].
- Signal strength – A closer node usually has a stronger signal than one far away [2] when similar transmission powers are used.
- Clock skew – Due to manufacturing reasons, each node has a unique clock skew, and the skew can be used to identify a node [6].

Techniques (e.g., those in [4 - 6]) relying on analysis of the physical radio signal typically require expensive hardware to obtain the necessary accuracy. However, the RSSI is a notable exception and the RSSI value is available in many wireless devices. On the other hand, RSSI is also unreliable for two reasons: 1) A common energy saving technique is to vary the transmission power to only the level needed for reaching the desired neighbor. If sensors dynamically change their transmission powers, the RSSI value itself is not very useful for node identification. 2) The signal strength of a transmission also varies due to environmental conditions, and can be unreliable even if the transmission power is fixed.

Faria and Cheriton [7] developed a RSSI based fingerprinting scheme, in which a fingerprint is the RSSI values recorded by multiple Access Points. Similarly to [2], they want to decide whether multiple transmissions came from the same source. In order to combat the effects of varying transmission power, the difference between RSSI readings from the same transmissions is used to determine an attacker. However, the actual differences between RSSI readings vary a lot and are not reliable. In our scheme, we use the fact that the ratio of RSSIs from two observers remains constant, regardless of the source transmission power. Yedavalli *et al.* [8] utilized RSSI for localization. The scheme in [8] is referred to as *Ecolocation*, and it is based on the distance-based rank-ordering by detector nodes with known locations. The assumption is that RSSI is correlated with distance, and the rank-ordering is determined by the location of the unknown node.

III. Wireless Fingerprinting Framework

A. Network Model

After sensor deployment, clusters are formed in a HSN. An efficient cluster formation scheme for HSNs can be found in [9]. Each cluster contains one H-sensor and a number of L-sensors, and the H-sensor is the cluster head. L-sensors respond to queries from and send data to its cluster head. A cluster head (H-sensor) aggregates data and then send it to the base station. An H-sensor is a more powerful node, and can communicate directly to all (or most) L-sensors in its cluster. L-sensors are small, low-power nodes and they send data to the cluster head via multi-hop communications.

B. Attack Model

An L-sensor may be captured and compromised, and then all the data, software and security materials will be revealed. Zhu *et al.* [10] suggested that there is a minimum time for an adversary to compromise a sensor node, and within the time period the network is assumed to be secure. In this paper, we make the same assumption as [10]. In addition, we assume that H-sensors are trustworthy. For example, H-sensors may be installed with tamper-resistant hardware. This is a reasonable assumption for powerful H-sensors.

Our main goal is to identify the source of a malicious collision attack, where an adversarial node makes transmissions timed to cause collisions with legitimate neighbor communications, for the purpose of disrupting traffic. For example, suppose the IEEE 802.11 MAC is used, and node *u* wants to send a packet to a neighbor node *v*. Based on the RTS/CTS exchanges, a neighboring adversarial node *x* knows the timing of *u* to transmit the data packet, and *x* can transmit a noise that overlaps with the data packet and hence cause collisions. The malicious collision attack also allows an attacker to carry out the selective forwarding attack [1] for routes that it isn't actually on. Since the attacker may not follow any protocol, we make no assumptions about the format of the collision packet other than that it has measurable signal strength to all neighbors.

C. Building the Fingerprints

In this paper, we propose a scheme that can identify the source of a malicious collision attack by using the RSSI readings. The scheme is based on the fact that the ratios of the RSSI readings between two neighbor nodes remain the same (or very close) for different transmissions from the same source, even the transmissions use different powers. Denote $\mathrm{R}_u(1)$ as the RSSI reading at node *u* for transmission #1. Suppose that neighbor nodes *u* and *v* record the RSSIs from two transmissions, if the following result holds:

$$\mathrm{R}_u(1)/\mathrm{R}_v(1) \approx \mathrm{R}_u(2)/\mathrm{R}_v(2) \qquad (1)$$

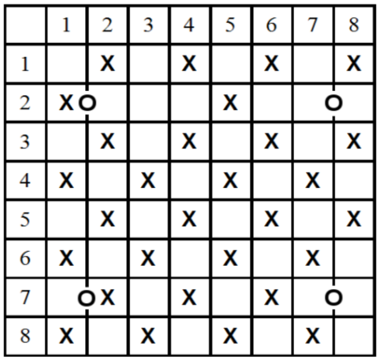
then we can claim that the same source node transmitted packet #1 and #2.

The RSSI ratio in equation (1) is a fingerprint of a node. In this subsection, we discuss how to build the fingerprints for the identification. During the initiation phase (assumed no attacks), all L-sensors send *hello* messages with a sequence number by using the same power. Each L-sensor records the RSSIs of neighbors' *hello* messages and the corresponding sequence numbers. Then the RSSI values are sent to the cluster head (denoted as H).

A sample set of RSSI readings [11] is given in Table I. These RSSI readings are data from actually wireless 802.11 transmissions experimented by the Orbit test-bed [12] at Dartmouth College. The experiment layout is shown in Figure 1, where 29 nodes are deployed in a grid of 8x8 cells, and the cell length is 1 meter. Xs denote nodes, and Os denote noise generators. The node is labeled by its coordinates in the grid, e.g., 1-2 is the node locates at row 1 and column 2. In Table I, the first row is the node label *x-y*. Rows 2 – 6 list the RSSI readings of five *hello* messages sent by the node at location 1-2, as recorded by all of its neighbors.

TABLE I. RSSI OF TRANSMISSIONS FROM NODE 1-2

	1-2	1-4	1-6	1-8	2-1	2-5	3-2	3-4	3-6	3-8	4-1
r-1	-1	38	19	21	23	27	35	25	20	22	26
r-2	-1	38	18	20	21	26	34	24	18	22	26
r-3	-1	37	14	18	19	24	30	22	16	19	23
r-4	-1	37	12	16	16	20	32	20	14	16	21
r-5	-1	36	11	15	16	21	32	20	13	16	20
ave	-1	37.2	14.8	18	19	23.6	32.6	22.2	16.2	19	23.2



	1	2	3	4	5	6	7	8
1		X		X		X		X
2	X O				X		O	
3		X		X		X		X
4	X		X		X		X	
5		X		X		X		X
6	X		X		X		X	
7	O	X		X		X	O	
8	X		X		X		X	

Figure 1. The network topology

The set of RSSIs for a given transmission from multiple neighbors are referred to as a report. For example, in Table I, report 1 includes all RSSIs in row 2. When a node logs a RSSI, the timestamp is recorded as well. The timestamp is sent to H along with the RSSI value. The fingerprint of a node is the set of all reports corresponding to *hello* messages sent by that node.

To reduce the overhead of RSSI fingerprint generation, the following schemes may be used:

1) If the IEEE 802.11 MAC is used, then the RTS/CTS packets can be used for recording the RSSIs and hence generating the fingerprint of a node.
2) If there are no MAC control packets being sent before the data packet (e.g., TDMA is used), then the RSSI can be obtained from the packet header. The header should be received by all neighbors such that a neighbor node knows whether this packet is intended for itself.

In both cases, no dedicated packets (e.g., *hello* messages) are used to generate the RSSI fingerprints, and hence the communication overhead is reduced.

After receiving all the reports, the cluster head H will compare the reports based on the timestamp to make sure RSSIs from the transmission is used to build the fingerprint. When a collision attack happens, there are two transmissions (the legitimate packet and the collision packet) occurring simultaneously. Suppose node *u* transmits to *v*, and node *x* causes a collision. When node *v* detects a collision, it assumes that there is a collision attack. However, *v* does not know who the attacker is. After detecting a collision attack, node *v* sends to all of its 2-hop neighbors an *alarm* message, which includes the legitimate sender ID *u* and the time *t* of the collision attack. Each 1-hop neighbor of node *u* should hear the legitimate transmission from *u* (or the collision). Hence, *u*'s 1-hop neighbors will not response to the *alarm* message (i.e., do NOT report to H). When other nodes receive the *alarm* message, each sends to the cluster head H a *report* message that includes the RSSI and the timestamp of a transmission around the time *t*. H will use the timestamps to correlate the readings. After collecting the RSSI readings from the *report* messages, H will build a RSSI ratio, and compare it with the RSSI ratio fingerprint. The node that has the closest match is considered as the attacker. We discuss the details of several identification schemes in Section VI.

IV. Effective schemes for identifying the attacker

A. The Average RSSI Value Scheme

To reduce the communication overhead of building fingerprints, each L-sensor should aggregate the RSSI readings from all its neighbors and only send a single report to H. A simple way to do this is for each L-sensor to take the average RSSI of multiple *hello* messages and send the average instead of the individual RSSIs. To minimize the variations of RSSI readings, *hello* messages are transmitted with a constant power. The above scheme is referred to as the Average RSSI Value (ARV) scheme. For example, for the RSSI data in Table I, each L-sensor computes the average of the RSSI values, as listed in the last row, and sends the average to H.

When a collision attack happens, H collects the event reports from L-sensors, and builds the RSSI ratio of the attacker. Then H compares the attacker's RSSI ratio with that of each candidate node y (neighbors of node v). A score is used to indicate the magnitude of the difference between the RSSI ratios. The RSSI ratios are computed for every pair of nodes i and j that have valid RSSI readings stored in the fingerprint and are listed the event reports. A candidate y's score is the average of the differences of RSSI ratios for all nodes i and j. The candidate y with the lowest score is identified as the source of the collision attack. Figure 2 lists the ARV scheme. In Figure 2, Report is the event report being analyzed, FPy is the fingerprint of node y, and i, j, y are L-sensors.

```
1 foreach candidate node y
2   score[y] := ∞
3   foreach iϵ(FPy and Report)
4     scorej := 0,    n := 0
5     foreach jϵ(FPy and Report), j≠i
6       $scorej \mathrel{+}= \frac{\text{Report}[j]}{\text{Report}[i]} - \frac{\text{FPy}[j]}{\text{FPy}[i]}$
7       n := n + 1
8     scorej := scorej / n
9     if scorej < score[y] then  score[y] = scorej
```

Figure 2. The ARV scheme

B. The Constraint-based Average RSSI Value Scheme

Ecolocation [8] uses the concept of constraints to estimate a node's position. In [8], a constraint is given by the distance-based rank ordering of a pair of neighbors. I.e., if neighbor *i* is closer than neighbor *j* to a node *u*, then the **constraint match** requires that the distance between *u* and *i* is less than that between *u* and *j*. In [8], a set of constraints is used to estimate the location of a node.

We apply the constraint technique to the ARV scheme, and refer to this new scheme as Constraint-based Average RSSI Value (CARV) scheme. During network initiation, each L-sensor collects RSSI readings from their neighbors, and sends the average RSSI to its cluster head H. When a collision attack is detected, neighbor L-sensors send RSSI reports to H for analysis.

We define a constraint function $c(s, t)$ as follows:

$$c(s,t) = \begin{bmatrix} -1 \ \ if \ \ (s<t) \\ 0 \ \ if \ \ (s=t) \\ 1 \ \ if \ \ (s>t) \end{bmatrix} \qquad (2)$$

Basically, the constraint function $c(s, t)$ compares two input values s and t and determines which is larger. A constraint is matched if and only if for two pairs of RSSI values (s_1, t_1) and (s_2, t_2), the following holds:

$$c(s_1, t_1) = c(s_2, t_2) \qquad (3)$$

```
1 foreach candidate node y
2   score[y] := 0
3   foreach iϵ(FPy and Report)
4     foreach jϵ(FPy and Report), j>i
5       if c(FPy[i], FPy[j])=c(Report[i], Report[j])
6         then  score[y] := score[y] + 1
7         else score[y] := score[y] – 1
```

Figure 3. The constraint based average RSSI value scheme

Constraints are calculated once for every pair of nodes *i* and *j* that have RSSI readings for each candidate node *y* in both the fingerprint and the *event report*. For each candidate node *y*, H computes a **score** that is the number of matched constraints minus the number of violated constraints. The candidate node *y* with the highest score is selected to be the best match. Figure 3 shows the algorithm.

C. The Hybrid Scheme

The ARV scheme is based on the fact that the RSSI ratio between two detector nodes for a given node should remain the same. The CARV scheme relaxes this requirement, and only considers if the one RSSI is larger than the other, instead of considering the actual ratio value. In this subsection, we present a Hybrid Average RSSI Value (HARV) scheme. The HARV scheme is similar to CARV scheme, but we change the way constraints are verified. As shown in Figure 4, only line 5 of the scheme is different from CARV, where a constraint is matched if the difference of two RSSI ratios is less than a threshold ε. In [2], a threshold of 5σ was used in the Sybil attack detection experiments, where σ is the Standard Deviation of the difference in RSSI ratios of consecutive messages as recorded by two detector nodes. In their experiments, 5σ = 0.5. We conducted experiments by varying ε, and found 0.5 to be a good value. The results shown below were collected with the parameter $\varepsilon = 0.5$. Again, the candidate node *y* with the highest score is selected as the source of the attack.

```
1 foreach candidate node y
2   score[y] := 0
3   foreach iϵ(FPy and Report)
4     foreach jϵ(FPy and Report), j>i
5       if $\left|\frac{\text{Report}[i]}{\text{Report}[j]} - \frac{\text{FPy}[i]}{\text{FPy}[j]}\right| < \boldsymbol{\varepsilon}$
6         then  score[y] := score[y] + 1
7         else score[y] := score[y] – 1
```

Figure 4. The hybrid scheme

D. The Localization-based Scheme

In this subsection, we present a Localization-based (LOC) scheme. The LOC scheme is an implementation of the location estimation scheme described in [13]. If each L-sensor knows its own location, then it is possible to estimate the location of the source of any transmission that is observed by at least four L-sensors. Basically, we can estimate the location by solving the following equation for *x* and *y*:

$$\begin{aligned} (x-x_i)^2 + (y-y_i)^2 &= \left(\frac{R_i}{R_j}\right)^{\left(\frac{1}{\alpha}\right)} \left((x-x_j)^2 + (y-y_j)^2\right) \\ (x-x_i)^2 + (y-y_i)^2 &= \left(\frac{R_i}{R_k}\right)^{\left(\frac{1}{\alpha}\right)} \left((x-x_k)^2 + (y-y_k)^2\right) \qquad (4) \\ (x-x_i)^2 + (y-y_i)^2 &= \left(\frac{R_i}{R_l}\right)^{\left(\frac{1}{\alpha}\right)} \left((x-x_l)^2 + (y-y_l)^2\right) \end{aligned}$$

where R_i is the RSSI recorded by node *i; i, j, k,* and *l* are the L-sensors that observed the transmission made by the node located at (x, y); and α is the distance-power gradient.

H obtains the R_i from *event reports*, and then derives the location of the source node based on Equation (4). The L-sensor closest to the source location is considered to be the attacker. One distinct advantage of this scheme is that it does not rely on any fingerprint. The location is derived based on a single, independent transmission. Also, it is not the identity of the node that is revealed, but the location.

V. Performance Evaluation

In this Section, we present the performance evaluation of the four schemes in Section IV. We utilize the RSSI data collected by Kaul *et al.* [11] on the ORBIT test bed [12]. The test-bed topology is shown in Figure 1, where 29 nodes were deployed in a grid of 8x8 cells with 1 meter cell size.

Five sets of RSSI data were collected, varying the power of the noise from -20 dbm to 0 dbm in an increment of 5 dbm. Each node made 300 transmissions; the RSSI for each transmission was recorded by the remaining nodes. We use the first 100 transmissions as our training set, and the remaining 200 transmissions as our data set. In our evaluations, 20 data sets were used for test purpose. The 20 transmissions were chosen by taking every 10th message from transmission 100 to 300. A close inspection of the generated fingerprints shows that the data set includes one node that was failed. That is, there are no valid readings for that node. Further, there are three nodes that have poor quality fingerprints. These four nodes cause the vast majority of the inaccuracy in the test.

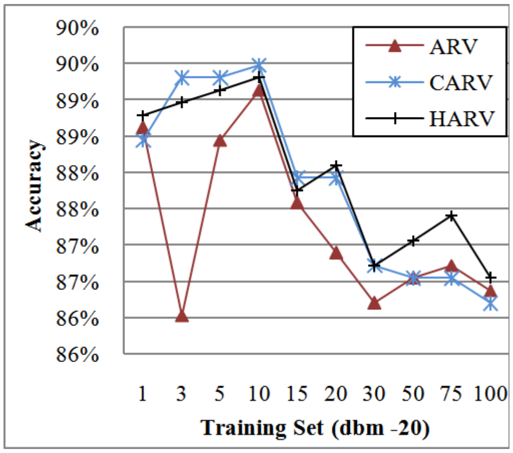


Figure 5. Identification accuracy vs. training set size

A. Finding the Optimal Size of Training Set

A parser was written in C++ to build the fingerprints and analyze the data against the test set. In order to compare the various schemes discussed in Section III, trials were run by varying the size of the training set. A training set of size *n* used the first *n* messages for training. The test set remained the same for comparison purpose. Figure 5 plots the accuracy of identifying the attacker Vs the size of the training set, and it shows that the training set size does NOT have significant impact on the accuracy. Note the accuracy varies between 0.86 and 0.89.

B. Evaluation of the LOC Scheme

Since the LOC scheme does not rely on RSSI fingerprints, it is considered different from the other three schemes. The algorithm used to solve Equation (4) is given in Figure 6, and is explained below: First, we rearrange the equations and move the terms to one side. We seek to find values for *x* and *y* that minimize the error. Four detector nodes *i, j, k, l* are chosen at random, and α is set to 2. The locations of the four detector nodes are given as (x_i, y_i), (x_j, y_j), (x_k, y_k), (x_l, y_l), respectively. For each transmission, the algorithm is run to estimate the sender's location, and the L-sensor closest to the computed location (in terms of Euclidean distance) is considered as the sender.

```
$minErr := \infty$
for $x = 0$ to $10$ step $0.1$
    for $y = 0$ to $10$ step $0.1$
        $err := \left|(x-x_i)^2 + (y-y_i)^2 - \left(\frac{R_i}{R_j}\right)^{\left(\frac{1}{\alpha}\right)}\left((x-x_j)^2 + (y-y_j)^2\right)\right|$
            $+ \left|(x-x_i)^2 + (y-y_i)^2 - \left(\frac{R_i}{R_k}\right)^{\left(\frac{1}{\alpha}\right)}\left((x-x_k)^2 + (y-y_k)^2\right)\right|$
            $+ \left|(x-x_i)^2 + (y-y_i)^2 - \left(\frac{R_i}{R_l}\right)^{\left(\frac{1}{\alpha}\right)}\left((x-x_l)^2 + (y-y_l)^2\right)\right|$
        if $err < minErr$
            then $bestX := x$, $bestY := y$, $minErr := err$
```

Figure 6. The algorithm for computing the location

TABLE II. LOCALIZATION ERRORS

Dataset	Error (meter)	Standard Deviation
dbm -20	3.88	1.89
dbm -15	3.75	1.82
dbm -10	3.85	1.88
dbm -5	3.48	1.87
dbm -0	3.18	1.87

As shown in Figure 7, the performance of the LOC scheme is quite poor. In order to better understand why the performance was poor, we examine the average error of the resulting coordinates versus the actual coordinates of the source for each transmission. The results are listed in Table II. For all data sets, the average localization error was greater than 3 meters. This is consistent with the results in [14], which found that a median error of 10 feet (3 meters) can be expected with localization based on IEEE 802.11. devices. Given that the nodes are placed on a 1-meter grid, an average error of 3 meters renders the LOC scheme useless.

C. The Accuracy of the Schemes

We evaluate the accuracy of the four schemes under different noise levels. The accuracy is defined as the percentage of a scheme correctly identifying the source node of a transmission. Specifically, we tested the accuracy of the four schemes under five different ambient (background) noise levels, from -20 dbm to 0 dbm, with an increase of 5dbm. The results are reported in Figure 7. As we can see the HARV performs better than other schemes, especially when the noise level increases. HARV has accuracy between 0.8 and 0.9. ARV and CARV perform reasonably well, but degrade a little bit when noise level increases. LOC performed poorly in all cases.

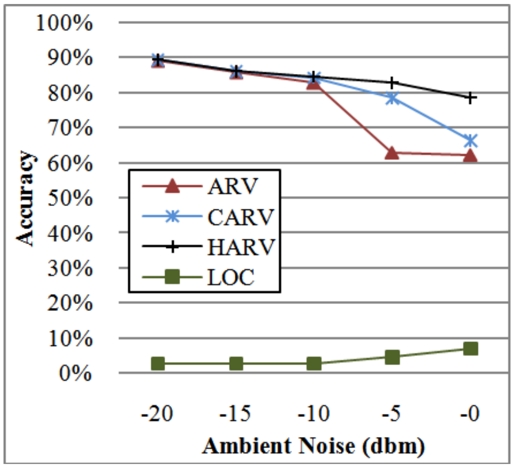


Figure 7. Identification accuracy vs. ambient noise level

VI. Conclusion

In this paper, we studied the malicious collision attack in wireless sensor networks. The attack can be easily launched by a compromised or hostile node by timing its transmission of a short noise and cause a collision with neighbor's transmission. This attack does not consume much energy of the attacker but can seriously disrupt communications in the network. Due to the wireless broadcast nature, it is not trivial to identify the attacker. In this paper, we proposed three effective schemes (ARV, CARV, and HARV) for identifying the source of the collision attack. The schemes only require low-cost hardware and very suitable for small sensor nodes. The schemes are based on the physical-layer Received Signal Strength Index (RSSI) readings and utilized the fact that the ratio of RSSIs from two neighbors is consistent for the same send. One of the schemes – the HARV scheme degrades gracefully as the ambient noise increases. We evaluated the performance of the schemes based on RSSI data collected from real wireless transmissions. Our results showed that the three schemes can correctly identify the source of a collision attack with greater than 85% accuracy. Our results also showed that the traditional localization scheme performed poorly.

Acknowledgment

This research was supported in part by the US National Science Foundation (NSF) grants CNS-0963578, CNS-1002974 and CNS-1022552, as well as the US Army Research Office under grant W911NF-08-1-0334.

References

[1] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 113-127, 2003.

[2] M. Demirbas and Y. Song, "An RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks," *Proc. of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 564-570, 2006.

[3] Y. Law, L. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-Efficient Link-Layer Jamming Attacks Against Wireless Sensor network MAC Protocols," *Proc. of the 3rd ACM Workshop on Security of Ad hoc and Sensor Networks*, pp. 76-88, Nov. 2005.

[4] D. Loh, C. Cho, C. Tan, and R. Lee, "Identifying Unique Devices through Wireless Fingerprinting," *Proc. of the First ACM Conference on Wireless Network Security*, pp. 46-5 , 2008.

[5] J. Hall, M. Barbeau, and E. Kranakis, "Detection of Transient in Radio Frequency Fingerprinting Using Signal Phase," *Wireless and Optical Communications*, ACTA Press, pp. 13-18, 2003.

[6] T. Kohno, A. Broido, and K. Claffy, "Remote Physical Device Fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93-108, Apr-Jun, 2005.

[7] D. Faria and D. Cheriton, "Detecting Identity-Based Attacks in Wireless Networks Using Signalprints," *Proc. of the 5th ACM Workshop on Wireless Security*, pp. 43-52, 2006.

[8] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: A Sequence Based Technique for RF Localization in Wireless Sensor Networks," *Proc. of the 4th International Symposium on Information Processing in Sensor Networks*, pp. 285-292, 2005.

[9] X. Du and F. Lin, "Maintaining Differentiated Coverage in Heterogeneous Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 5, issue 4, pp. 565–572, Sept. 2005.

[10] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," *Proc. of the 10th ACM Conference on Computer and Communications Security*, October 2003, pp. 62-72.S.

[11] Kaul, Gruteser, and I. Seskar, "CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth," accessed Feb. 23, 2009, http://crawdad.cs.dartmouth.edu/meta.php?name=rutgers/noise.

[12] ORBIT Emulator, http://www.orbit-lab.org/, accessed 23 Feb 2009.

[13] S. Zhong, L. Li, Y. Liu, and Y. Yang, "Privacy-Preserving Location-based Services for mobile users in Wireless Networks," Yale Computer Science, Tech. Rep. YALEU/DCS/TR-1297, July 2004.

[14] E. Elnahrawy, X. Li, and R. Martin, "The Limits of Localization Using Signal Strength: A Comparative Study," *Proc. of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, Oct. 2004, pp. 406-414.