Fast Jamming Detection in Sensor Networks

Kartik Siddhabathula, Qi Dong, Donggang Liu and Matthew Wright
Department of Computer Science
University of Texas at Arlington
{kartik.siddhabathula, qi.dong}@mavs.uta.edu, {dliu, mwright}@cse.uta.edu

***Abstract*—Wireless Sensor Networks (WSN) are vulnerable to jamming attacks where an adversary injects strong noises to interfere with the normal transmission. It is crucial to detect such jamming attacks as fast as possible. Existing studies have shown that an effective indicator of jamming is the packet delivery ratio (PDR). However, current PDR-based schemes use the end-to-end packet delivery ratio, which requires one to observe communication for a long time before a good decision is made. In this paper, we propose *collaborative detection*, which evaluates the packet delivery ratio in an given area instead of a pair of nodes. The intuition is that the attacker often jams an area of his interest, not just two specific nodes. The benefit is that we can detect jamming attacks in a much faster way. We have evaluated the performance of our idea on TelosB motes. The results show that we can effectively and quickly detect jamming attacks.**

I. Introduction

Wireless sensor networks consist of autonomous sensors to monitor the conditions in an area of interest and report their observations to a base station for further analysis. They have become a useful tool in civilian, industrial and scientific applications such as border security, land slide detection, green house monitoring. In hostile environments like border security, the security of the sensor network has to be ensured.

Due to the shared medium of communication, wireless sensor networks are vulnerable to Denial of Service (DoS) attacks [2], e.g., the *jamming attacks*. A jammer hampers the communication between benign nodes by attacking either the network layer, the physical layer, or the Medium Access Control (MAC) layer. In this paper, we focus on the MAC layer jammers for wireless communication medium. A MAC layer jammer does not adhere to the MAC protocol and emits a radio signal which interferes with the normal working of the network and causes a DoS attack. Depending on the power of jammers, it can either cause a part of the network to fail or even bring down the whole network.

Jamming causes many problems for real world applications. For example, in border security, an intruder can jam the communication and cross the border without being detected. Thus, in hostile environments, it is essential to be able to detect the place where the channel is jammed [9], [20], [18] or deliver the messages out of the jammed area [19], [3], [9], [1].

In this paper, we focus on methods to detect the place where jamming attacks are launched. To achieve this, we need to first detect the jamming attacks. Once we notice that the communication is jammed, we can send an alert to the base station. Certainly, a node inside a jammed area may not be able to send its alerts out since the packet reception of its neighbors may have already been disrupted. However, the sensor nodes on the edge of the jammed area can often get their alerts out since some of their neighbors are outside of the jammed area [18]; these nodes will be the ones who detect jamming attacks and also report alerts to the base station.

The remaining problem is thus how to quickly detect jamming attacks. Previous studies have shown that the nodes being jammed will see a substantial drop in the packet delivery rate (PDR) [20]. Hence, once a node realizes that its packet delivery ratio drops significantly, a jamming alert can be produced. However, current PDR-based schemes use the end-to-end packet delivery ratio, which requires one to observe communication for a long time before any good decision is made. In this paper, we propose a *collaborative detection scheme*. The main idea is to evaluate the packet delivery ratio in an area instead of pairs of nodes since the attacker usually jams the area of his interest, not just the communication between some specific pairs of nodes. In other words, we use observations from other nodes to speed up the jamming detection. We have evaluated the performance of our protocol on TelosB motes. The results show that we can effectively and quickly detect jamming attacks.

This paper is organized as follows. The next section outlines the related work. In Section III we discuss the network and attack models. In Section IV we give a description of the protocol. In Section V we evaluate the proposed scheme. In Section VI we present the experimental results. In Section VII we give our conclusion and some future work.

II. Related Work

Detecting jamming attacks in wireless networks has attracted a lot of attention recently [9], [20], [18]. In [9], the authors proposed to build a model for the signal collision rate when there are no attacks. Such model is then used to detect if the network is under jamming attacks in hostile situations. The paper provided a theoretical study on the trade-off between the detection time, detection rate, and false alarm rate. In our paper, we use real experiments to study the performance of jamming detection. In [20], the authors studied various jamming attacks on WSN and proposed a detection method based on Received Signal Strength Indicator (RSSI) and Packet Delivery Ratio (PDR). However, their protocol requires a sensor node to observe the communication for many rounds before a good decision is made. In our protocol, we use the observations from multiple nodes to speed up the detection. In [18], the authors proposed to locate the jammed area once

the jamming attacks are detected. They assumed the existence of a jamming detection technique. In our paper, the focus is on how to detect such attacks quickly. Several papers have studied how to efficiently jam sensor networks. In [21], the authors proposed an efficient jamming attacker model for situations when the attacker has no knowledge about the target protocol. In [9], the authors described jamming strategies that are easy to launch but difficult to detect.

III. Network and Attack Model

We consider the network to be a static network, i.e., sensor nodes do not change their locations after deployment. All the nodes have limited supply of energy. We assume that the communication in the network is protected by the shared keys between sensor nodes. Many existing key establishment schemes [13], [5], [10] can be used for such purpose. We do not consider compromised nodes in the network.

We assume that the attacker's goal is to disable the network function for a period that is long enough to accomplish a task, e.g., crossing the border without being caught. As a result, the attacker will need to interrupt the communication dramatically so that no or very few messages can get out of the jammed area, e.g., the path for crossing the border. In addition, the attacker has to jam the area that is large enough to cover all possible nodes that may receive the reports from the nodes that sense the intruders.

Due to the above reason, we believe that it is more interesting to quickly detect jamming attacks that substantially reduce the packet delivery rate. One example is the so-called *constant jammer* [20]. This type of jammers continuously emits radio signals to cause jamming. In this paper, we have implemented a constant jammer using a TelosB mote that uses a CC2420 radio chip. We implemented it by disabling the Clear Channel Assessment (CCA) bit. Our jammer continuously sends out a packet irrespective of the activity on the channel. Note that although our experiment is based on a constant jammer, our method works on any jammer that substantially reduces the packet delivery rate.

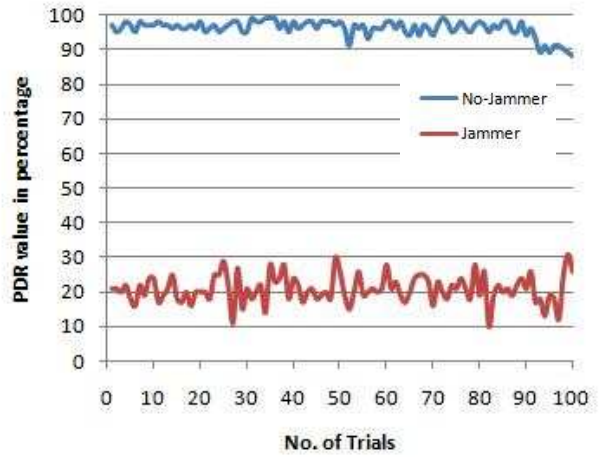

Fig. 1. Influence of jammers on PDR

Figure 1 shows the influence of our constant jammer on the communication between sensor nodes. The experiment focuses on the reception of messages. In the experiment, three nodes (a jammer, a receiver, and a sender) were placed in a straight line with a distance of 10 inches from each other. The receiver was placed in the middle. We carried out 100 trials with the receiver node being under attack and another 100 trials when the jammer is replaced with a normal node sending out messages continuously to simulate the signal collision that happens frequently in benign situations. In each trial, the sender sent out 100 packets while the receiver node kept track of the number of packets received. In the figure, the X-axis represents the index of the trial and the Y-axis represents the number of times the PDR was found to have that value in the corresponding trial. Note that PDR here is the packet reception rate for a receiver node. From the figure we can see that there is a clear distinction in the number of packets a node receives with and without jamming attacks.

IV. The Proposed Protocol

Our intuition is that an observation of packet loss over an area leads to decide regarding the existence of jammer faster when compared to observing the packet loss between a pair of nodes. Specifically, we divide the time line into multiple intervals and have sensor nodes periodically send out beacon signals. We then observe the loss of messages within each time interval for jamming detection. From Figure 1, we know that there is a huge difference in the PDRs with and without jamming attacks. As a result, for a given time interval, if a node sees a significant drop on the number of beacons received from neighbors when compared to what was observed in the last time interval, we know that this node is jammed. Certainly, frequently broadcasting beacon messages allows us to detect jamming faster but consumes more energy and drains out the batteries of the nodes faster. Hence, the length of time intervals have to be adjusted to make a good trade-off between the detection time and the energy consumption.

A. Decision Process

Each node maintains two arrays which (for the ease of understanding) we will call as $Current$ and $History$. $Current$ contains the list of the nodes whose beacons was received in the current time interval and the array $History$ contains the list of the nodes whose beacon was received in the previous time interval. Each element in the array is either 0 or 1. 0 means that the beacon of the corresponding node has not been received, and 1 means that the beacon of the corresponding node has been received. Both arrays are initialized to all 0s. We assume that sensor nodes are organized into clusters. This can be achieved by using some existing clustering algorithms [11], [15]. Each cluster conducts jamming detection and reports at most one alert in case of jamming.

The protocol carried out by each node in a cluster are as follows. First, each node in a cluster broadcasts a beacon that carries its ID repeatedly at each time interval. This helps the nodes which receive this beacon to keep track of the number

of beacons they are able to receive within this time interval. Each node broadcasts it's beacon only once in each interval.

Second, when a sensor node receives a beacon message, it marks the corresponding bit in the array $Current$ as 1, meaning that it has received the beacon from the corresponding cluster member. At the end of each time interval, every sensor node carries out a check to see whether there is jamming in the network. If yes, an alert will be send to the base station. The decision regarding the existence of jammers is made according to the following: Let X_n denotes the number of nodes whose beacon was received during the time interval t_n. An alert is raised if $X_n < \tau \times X_{n-1}$, where τ is a threshold. The alert will be send to the base station δ times to ensure that the base station receives it with a high probability.

At the end of each time interval, the content of array $Current$ is copied into the array $History$ and the array $Current$ is then re-initialized to 0.

B. Threshold selection

The decision regarding the existence of jammer is based on the comparison between the number of beacons received by a node in two consecutive time intervals. A threshold τ is used in such comparison. Certainly, the selection of a good τ is important. In this paper, τ is configured empirically, i.e., through a set of real experiments. We use the results in Figure 1. τ is assigned a value which lies between the two distributions. In this paper, we simply set τ to 0.55, which clearly separates the two distributions.

V. Theoretical Analysis

In this section we will analyze the detection rate, false alarm rate, and detection time. For the sake of presentation, we list some important notations used in our analysis:

- N: Total number of nodes in a cluster.
- M: Total Number of jammed nodes in a cluster.
- p: Probability that a message is lost due to channel noise or packet collision.
- δ: Number of alert messages sent by a jammed node.

The detection of jamming will be done whenever a jammed node is able to send an alert message out of the jammed area and that message is successfully received by at least one non-jammed node. There will be no detection if all the alert messages are lost, which can happen if all the nodes in the cluster and their neighbors are getting jammed. In this case, we will resort to other clusters to do the detection. Thus, for simplicity, we assume that the cluster is not completely jammed, and at least one unjammed node is located within the radio range of each jammed node. Given that the probability that a message is lost is p, the number of alert messages sent out by a node in case of detection is δ and the number of jammed nodes is M, the total number of alert messages generated in case of jamming detection are $M \times \delta$. The probability that all the alert messages are lost is thus at most $p^{M\times\delta}$. Therefore the detection rate R_D can be estimated by $R_D = 1-p^{M\times\delta}$. For example, if $M = 2$, $\delta = 5$, and $p = 0.3$, then the probability of detection is about 0.99999. In general, δ should be large enough to give us a good trade-off between energy consumption and detection. We can also see that the length of time intervals has no impact on the detection rate.

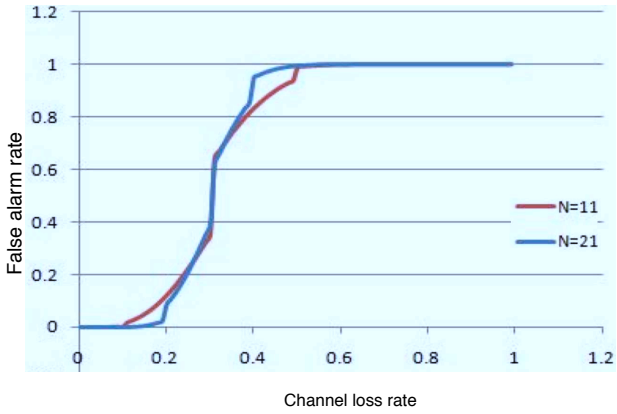

Fig. 2. The false alarm rate

A false alarm is an alert raised by a node in the network without the presence of a jammer. Note that an alert should be raised if a node receives less than τ times the beacons it received in the previous time interval. If this happens when there are no jammers, the alert is a false alarm. Given that the number of nodes in a cluster is N, a node expects to receive beacons from $N-1$ nodes. If the channel loss rate is p, then that node will receive $(N-1)\times(1-p)$ beacons on average. The probability that a false alarm will be generated can thus be estimated by the probability of $X_n < \tau\times(N-1)\times(1-p)$. Therefore, the false alarm rate FA can be estimated by

$$FA = \sum_{i=(1-\tau)(N-1)(1-p)}^{N-1} \binom{N-1}{i} p^{((N-1)-i)}(1-p)^i.$$

Figure 2 shows the false alarm rate for different values of N. From the graph, we can see that increase in the channel loss rate results in a higher false alarm rate.

The detection time is defined as the time difference between the time when the attack was launched and when an alert was successfully received by a non-jammed node. In our protocol, the time at which a node receives a message is random as well as the time at which jammer launches attack. For simplicity, we assume that there are no packet losses without the presence of the jammer. If the length of the time interval is t time units, we divide it into two sub intervals. The first sub interval lasts for $\tau \times t$ time units, and the second sub interval lasts for $(1-\tau)\times t$ time units. If the attack is launched during the first sub interval, then the attack will be detected at the end of the current time interval, and if the attack is launched during the second sub-interval, then it will be detected at the end of the next time interval. Thus, the average time for detection at the end of current time interval will be $t-(t\times\tau)/2$ and the average time for detection completed at the end of next time interval will be $2t-(t\times(1-\tau))/2$. Therefore,the overall

average detection time for the protocol can be estimated by

$$(\tau \times (t - \frac{(t \times \tau)}{2}) + ((1 - \tau) \times (2t - (t \times \frac{(1 - \tau)}{2})).$$

This equation can be further reduced to

$$(t\tau) + \frac{(t\tau)}{2} - 2t\tau + \frac{(t\tau)}{2} - \frac{t\tau^2}{2} - \frac{t\tau^2}{2} + 2t - \frac{t}{2}.$$

Finally, the average detection time can be estimated by

$$\frac{3t}{2} - t \times \tau^2.$$

We can see that our scheme can detect jamming attacks after 1.5 rounds of transmission on average, which is much faster than the schemes in [20] where a node has to observe many (e.g., 10) rounds of transmission to make a good decision.

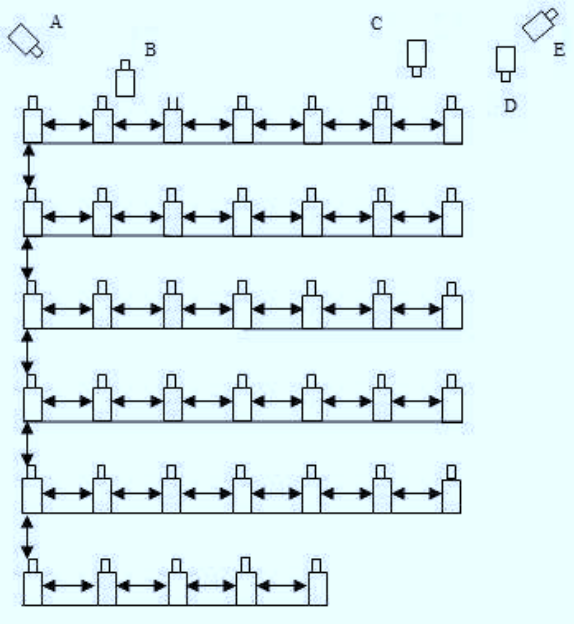

Fig. 3. Network layout

VI. Implementation and Experiments

The performance of our Protocol was tested on TelosB motes [4] running TinyOS [6]. TinyOS is a free and open source component-based operating system that targets embedded systems like sensor nodes. TinyOS applications are written in NesC, a dialect of the C programming language which has been optimized for the memory limitations of wireless sensor networks. TelosB motes use a CC2420 chipcon radio [16]. For our testing, the jammer and the benign motes transmit at the same power level. We assign the value of δ to be 10, i.e. each mote will send the alert message 10 times whenever it detects jamming in the network. Each alert message will be sent with a random time interval between 0 to 100 ms to reduce signal collision.

The experiment was carried out inside the lab in an area of 900 sq. inches. The protocol was tested against one constant jammer which was placed at various positions and at different orientation to jam different number of motes for each position and orientation of the jammer. The jammer was placed outside the perimeter of the network. The range of the TelosB motes is reduced in an indoor environment. We also further reduced their range by making them transmit at the lowest power level which is 1. The effective range of the motes transmitting at power level 1 was ranging between 15-25 inches. 40 motes which were running the protocol were arranged in a grid topology as it made for an easier deployment. Each mote was placed at a separation of 5 inches. The jammer was also programmed to transmit at the same power level as the motes which is power level 1. Figure 3 shows the network topology and the jammer's various position for testing the protocol. Depending on the position of a mote in the network, the number of neighbors for a mote varied from 6 to 14 motes. Though the motes were not time synchronized, they were all running at the same interval of time interval. Motes which lies on the edge of the network were having less number of neighbors and motes which lies inside the network were having more number of neighbors. Tests were carried out to record the detection time, detection rate, and the false alarm rate of the protocol.

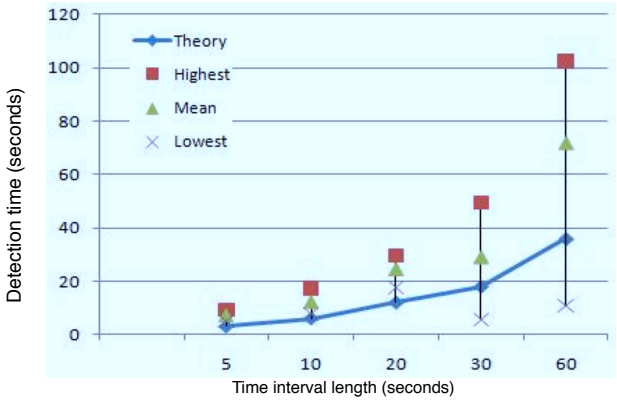

Fig. 4. Detection time

To calculate the detection time, one of the non-jammed mote was connected to the computer. Whenever that mote received an alert message it would print the reception time on the system. This gave us the time at which detection was complete. To get the time when the attack was launched, another mote was also connected to the computer which sent a message to the jammer on reception of which the jammer starts the attack on the network. That mote will print the time when the jammer is launching the attack on the system screen. By taking the difference between the two times we get the detection time. For the detection time, the length of time intervals are configured in five different ways, i.e., five sets of experiments are conducted, each having different interval length. Figure 4 shows the box plot for the detection time for five different time intervals which are 5 secs, 10 secs, 20 secs, 30 secs and 60 secs and a curve that represents the theoretical average detection time for each of the time interval length. From the figure, it is clear that, as the time interval increases the average

detection time increases. We can also see that the experiment results are consistent with the theoretical analysis result.

In Figure 3, the motes next to the letters are jammers, and each letter also indicates the position. When the jammer was placed at different position with different orientation, different number of motes were jammed. When the jammer was placed at position A, 16 motes were jammed; at position B, 12 motes were jammed; at position C, 20 motes were jammed; at position D, 8 motes were jammed; at position E, 6 motes were jammed. Figure 5 shows the detection rate of the protocol for each case. In experiments, 100 attacks were carried out for each position of the jammer. Our protocol achieved 100% detection rate when the jammer was placed at position E and B. The lowest detection rate was 97% in case of jammer being placed at position D. In the other two cases, i.e., when the jammer was placed at A and C, the detection rate is 98%.

Fig. 5. Detection rate

For the false alarm part, the motes were programmed to transmit a packet, other than their beacon packet every 993 millisecond with a certain probability. Transmitting a packet other than the beacon of the mote creates a situation where motes will be running multiple protocols. We wanted to see the impact of other protocols might have on our protocol. Additional packets by the motes increases the traffic of the network. With increased traffic, the chances of collision increases which results in loss of packets. The jammer was removed and only the 40 motes operated for 3 hours with a time interval of 10 seconds, i.e., a total of 1080 intervals. Interestingly, in all the cases with different probabilities for the additional packet being transmitted, only once was an alert raised in the network. Therefore, the false alarm rate of the protocol is only $\frac{1}{1080} \approx 0.0009$.

VII. Conclusion and Future Work

Wireless Sensor Networks are being widely used in various fields for different types of data collection and monitoring. They are a useful commodity in various civilian, industrial and scientific applications. Jamming attacks are one of the most popular attacks on WSN to cause a DoS. There is a need for the detection of these attacks quickly and accurately. In this paper we show through experiments that an observation of packet loss over an area gives us a faster detection of jamming attacks. From the experiment we also found that the protocol has a very low false alarm rate. In the future, we would like to test our protocol against different attacker models.

References

[1] G. Alnie and R. Simon. A multi-channel defense against jamming attacks in wireless sensor networks. In *ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 95–104, 2007.

[2] AUSCERT. Denial of Service Vulnerability in IEEE 802.11 Wireless Devices. http://www.auscert.org.au/render.html?it=4091.

[3] M. Cagalj, S. Capkun, and J.P. Hubaux. Wormhole-based antijamming techniques in sensor networks. *IEEE Transactions on Mobile Computing (TMC)*, 6, Jan 2007.

[4] Crossbow Technology Inc. Wireless sensor networks. http://www.xbow.com/Home/HomePage.aspx. Accessed in July 2009.

[5] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proc. ACM Conf. on Computer and Communications Security (CCS)*, November 2002.

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[7] Meng-Yen Hsieh, Yueh-Min Huang, and Han-Chieh Chao. Adaptive security design with malicious node detection in cluster-based sensor networks. *Comput. Commun.*, 30(11-12):2385–2400, 2007.

[8] Issa Khalil, Saurabh Bagchi, and Ness Shroff. Analysis and evaluation of secos, a protocol for energy efficient and secure communication in sensor networks. *Ad Hoc Networks.*, 5(3):360–391, 2007.

[9] M. Li, I. Koutsopoulos, and R. Poovendran. Optimal jamming attacks and network defense policies in wireless sensor networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1307–1315, May 2007.

[10] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *Proc. ACM Conference on Computer and Communications Security (CCS)*, October 2003.

[11] Donggang Liu. Resilient cluster formation for sensor networks. In *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS)*, page 40, 2007.

[12] Leonardo B. Oliveira, Adrian Ferreira, Marco A. Vilaça, Hao Chi Wong, Marshall Bern, Ricardo Dahab, and Antonio A. F. Loureiro. Secleach-on the security of clustered sensor networks. *Signal Process.*, 87(12):2882–2895, 2007.

[13] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security protocols for sensor networks. In *Proc. Intl. Conf. on Mobile Computing and Networks (MobiCom)*, July 2001.

[14] Michael Sirivianos, Dirk Westhoff, Frederik Armknecht, and Joao Girao. Non-manipulable aggregator node election for wireless sensor networks. In *ICST WiOpt*, January 2007.

[15] Kun Sun, Pai Peng, Peng Ning, and Cliff Wang. Secure distributed cluster formation in wireless sensor networks. In *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC)*, pages 131–140, 2006.

[16] Texas Instruments Inc. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. http://focus.ti.com/lit/ds/symlink/cc2420.pdf. Accessed in January 2008.

[17] Sudarshan Vasudevan, Brian DeCleene, Neil Immerman, Jim Kurose, and Don Towsley. Leader election algorithms for wireless ad hoc networks. In *Proceedings of DARPA Information Survivability Conference and Exposition*, pages 261– 272, 2003.

[18] A. Wood, J. Stankovic, and S. Son. JAM: A jammed-area mapping service for sensor networks. In *Proceedings of IEEE Real-Time Systems Symposium*, pages 286–297, 2003.

[19] W. Xu, W. Trappe, and Y. Zhang. Anti-jamming timing channels for wireless networks. In *WiSec '08: Proceedings of the first ACM conference on Wireless network security*, pages 203–213. ACM, 2008.

[20] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.

[21] Y.W.Law, L.V.Hoesel, J.Doumen, P.HartelL, and P.Havinga. Energy efficient link-layer jamming attacks against wireless sensor network mac protocols. In *Proceedings of ACM Workshop on Security of ad hoc and Sensor Networks (SASN)*, pages 76 – 88, 2005.