Self-propagate Mal-packets in Wireless Sensor Networks: Dynamics and Defense Implications

Bo Sun*, Dibesh Shrestha
Dept. of Computer Science
Lamar University
Beaumont, TX 77710
{bsun, dshrestha}@my.lamar.edu

Guanhua Yan
Information Sciences (CCS-3)
Los Alamos National Laboratory
Los Alamos, NM 87545
ghyan@lanl.gov

Yang Xiao*
Dept. of Computer Science
University of Alabama
Tuscaloosa, AL, 35487, USA
yangxiao@ieee.org

***Abstract*—In this paper, based on our proposed mal-packet self-propagation models in wireless sensor networks, we use TOSSIM to study their propagation dynamics. We also present a preliminary study of the feasibility of mal-packet defense in sensor networks. Specifically, based on random graph theory and percolation theory, we propose the immunization of the highly-connected nodes in order to partition the network into as many separate pieces as possible, thus preventing or slowing down the mal-packet propagation. We study the percolation threshold of different network densities and the effectiveness of immunization in terms of connection ratio, remaining link ratio, and distribution of component sizes. We also present an analysis of the distribution of component sizes.**

I. Introduction

Because of the low physical security, lack of resilience and robustness of underlying operating systems [1], and the ever increasing complexity of deployed applications, new system vulnerabilities keep being reported about Wireless Sensor Networks (WSNs). In [14], it has been demonstrated that self-propagate mal-packets can exploit memory-related vulnerabilities to propagate itself, thus taking over the whole network. Once the mal-packets start spreading by exploiting the monoculture of WSN applications, manual human intervention is hardly effective based on the past experience gained from defending against Internet worms. Therefore, self-propagate mal-packets have become an emergent threat towards information confidentiality, integrity, and service availability for WSNs.

In this paper, based on the implemented CSMA protocol in TOSSIM [15], we first study the propagation dynamics of mal-packets. The reason that we use TOSSIM is because it exploits the WSN domain and TinyOS design, and thus is more suitable for WSN research. We study the unicast and broadcast mal-packet propagation models and present their dynamics.

Existing works only focus on mal-packet propagation dynamics in sensor networks [14], [9], [10], [13]. Few works consider how to prevent such kind of propagation. Therefore, based on percolation theory and random graph theory, we further study the feasibility of defending against mal-packet propagation in WSNs. Specifically, we model the deployment of sensor nodes as a homogeneous spatial Poisson process in a two-dimensional space and study the effectiveness of immunizing some sensor nodes in order to protect WSNs. Immunizing a node means that the node cannot be infected by the self-propagate mal-packets. Therefore, our purpose is to know how to choose the approriate set of nodes in order to partition the WSN into as many separate pieces as possible, therefore preventing or slowing down the mal-packet propagation. Percolation theory [12] tells us that with the increase of immunized nodes, there exists a "critical phenomenon" at which the network suddenly becomes disintegrated. Intuitively, we select the most connected nodes to immunize. We study the impact of the immunization on the network topology in terms of connection ratio, remaining link ratio, and the distribution of separate component sizes. Simulation results demonstrate that the immunization can effectively prevent or slow down the large-scale outbreaks of mal-packets.

*This research was supported in part by the Texas Advanced Research Program under grant 003581-0006-2006 and the US National Science Foundation (NSF) under grants DUE-0633445, CNS-0716211, CNS-0737325, and CCF-0829827.

II. Mal-packet propagation model

In [13], we have presented a mal-packet propagation model based on 802.15.4 WSNs. The CC2420 radio of MicaZ motes is 802.15.4 compliant and presents development trends for low-power and low-voltage wireless applications. With the purpose of fast propagation, an infected node wants to exploit the channel in order to propagate as fast as possible. Based on these considerations, one example propagation model is illustrated in Fig. 1. A node which has a packet ready to send backs off for a random number of time between 0 and $2^{BE}-1$, where BE is set to 3 by default. If the channel is found to be busy again after the random backoff, BE increases by 1. This process is repeated until either BE equals *aMaxBE* (which has a default value of 5), at which point BE is frozen at *aMaxBE*, or until a certain maximum number of permitted random backoff stages, denoted as *macMaxCSMABackoffs*, is reached, at which point an access failure is declared to the upper layer.

The mal-packet exchange follows the pattern illustrated in Fig. 2. Each node can be in one of three states: *Invulnerable*, *Vulnerable and Infected*, and *Vulerable and Uninfected*. Based on whether existing applications are unicast or broadcast, mal-packet propagation can take different formats. For example, pairwise key or groupwise key mechanisms may be deployed

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

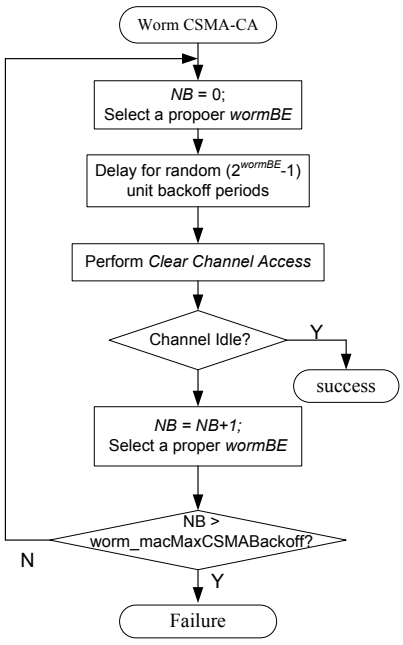


Fig. 1. Unslotted CSMA-CA for WSN Worms in 802.15.4 *nonbeacon-enabled* mode.

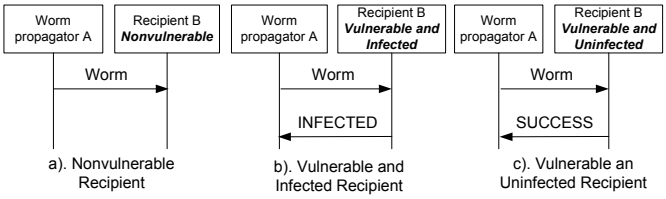


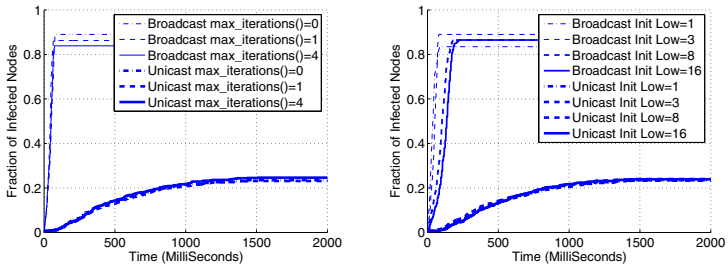
Fig. 2. Neighbor Feedback.

for different purposes. This may make mal-packet propagation adopt either unicast or broadcast approaches. We refer readers to [13] for details.

III. Mal-packet Propagation Dynamics

Unfortunately, the 802.15.4 standard has not been fully supported in TinyOS 2.x. Instead, a basic CSMA-CA algorithm is adopted (See *TossimPacketModelC.nc* for the implementation under TOSSIM). Its backoff mechanism is similar to those described in 802.15.4. Two important parameters play the same role as *NB* and *BE*. The *max_iterations()* value denotes the parameter *macMaxCSMABackoff* in 802.15.4, while the *init_low()* value denotes the lower bound of the backoff range. Therefore, based on TOSSIM, we observe the impacts of these two parameters on mal-packet propagations. Note that more details of *NB*, *BE*, *max_iterations()*, and *init_low()* can be found in [13]. We skip their introduction in this paper because of space limitations.

We randomly pick one node as an *infected* node. All other nodes are set to *vulnerable and uninfected*. Before the mal-packet begins propagation, every node uses a *Hello* protocol to discover its neighbors. We generate the uniform distribution



(a) Impact of *max_iterations()*. (b) Impact of *init_low()*.

Fig. 3. Impacts of Simulation Parameters.

of 700 nodes, 800 nodes, and 900 nodes in a 400m X 400m area and present its propagation dynamics.

A. Impacts of max_iterations()

In this set of simulation runs, we use the normal backoff values, and present the propagation dynamics under different max iterations for the 800 nodes in Fig. 3(a).

First, it is obvious that the broadcast propagation is much faster than that of the unicast. Also, broadcast propagation may infect much more number of nodes. The results are what we expect. Unicast propagation needs to transmit more mal-packets. However, the low transmission rates in WSNs may introduce more packet collisions. This will lead to more mal-packets being dropped by the network.

Second, even for broadcast propagation, it is still difficult to reach an 100 percent infection. WSNs suffer from very low transmission speed and unreliable transmission links. This may make packets prone to being dropped.

Third, given the broadcast propagation, the larger the value of *max_iterations()*, the higher the infection rate is. This is because a larger value of *max_iterations()* could give the nodes more chances to exploit the channel, thus increasing the successful transmissions of mal-packets. Note that in TOSSIM, a zero value of *max_iterations()* means infinity.

B. Impact of init_low()

In this set of simulation runs, we set the value of *max_iterations()* to 0, and present the propagation dynamics under different backoff values for the 800 nodes, as illustrated in Fig. 3(b). In Fig. 3(b), when the value of *init_low()* is 1, it means that the backoff value is set to *1 * init_low()*. Similarly, an *init_low()* of 3 means a backoff value of *3 * init_low()* is adopted.

Besides similar observations presented in Fig. 3(a), we can also see that, for the broadcast propagation, a smaller value of *init_low()* may make the propagation faster than that of a larger value of *init_low()*. A smaller *init_low()* value decreases the random waiting time when sensor nodes perform the backoff. Therefore, the mal-packet propagation speeds up.

C. Network Density

In this set of simulation runs, we adopt the default values of *max_iterations()* and backoff, and measure the impact of

network density on the propagation dynamics, as illustrated in Fig. 4.

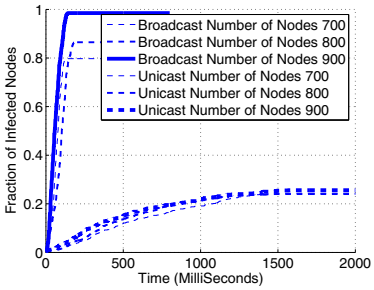
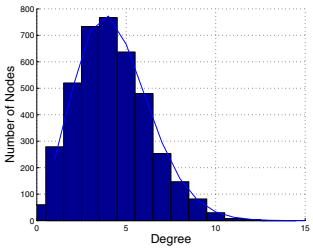
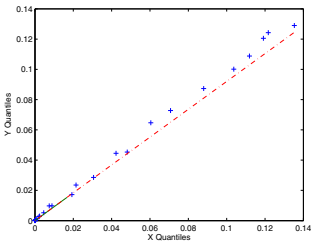


Fig. 4. Impact of Network Density.



(a) Deployment of Sensor nodes, fit with a Poisson Distribution.



(b) QQplot: Most of data points fall almost perfectly along the line, which is a good indicator that the degree is Poisson-distributed.

Fig. 5. Node Degree Distribution.

We observe that the denser the network, the higher the percentage of nodes is infected. Denser networks make nodes have higher node degrees and higher connectivity. This can facilitate the mal-packet propagation.

IV. Immunization-based Mal-packet Defense

We present an immunization-based mal-packet defense mechanism in this section. The following notations in Table I are used throughout the rest of the paper.

A. Network Model as Random Graph

TABLE I
NOTATIONS.

Symbol	Meaning
N	Number of sensor nodes deployed
r	Communication range of sensor nodes
λ	Node density
p_k	The probability that a randomly picked node has k neighbors

We use a homogeneous Poisson point process to model the distribution of sensor nodes. We assume that each node has a communication range of radius r. Therefore, two nodes are linked together if and only if they are not farther apart than a certain threshold. Consider N nodes that are uniformly distributed in a square area with side length of X. Therefore, for a node A, the number of nodes falling inside the circle around A, i.e., the number of neighbors of node A, is equal to $r^2\lambda$, where λ is the network density and is equal to $\frac{N}{X^2}$.

Under this assumption, we can use a random graph to model the deployment of sensor nods. p_k, the probability that a randomly picked node has k neighbors, is then equal to $\frac{\lambda^k}{k!}e^{-\lambda}, k = 0, 1, 2, \ldots$.

We perform a simple simulation to demonstrate this match. We simulate the deployment of $4,000$ nodes in a square area of $1,250 \times 1,250\ m^2$. Each node has a transmission radius of 25. We plot the degree distribution of these sensor nodes, as illustrated in Fig. 5(a). In Fig. 5(a), the solid line indicates the theoretical Poisson distribution.

Based on this simulation, we can calculate p_k. To illustrate whether p_k follows a Poisson distribution, we make a Quantile-Quantile plot (Q-Q plot) between a theoretical Poisson distribution and p_k, as illustrated in Fig. 5(b). As seen in Fig. 5(b), most of data points fall almost perfectly along the line, which is a good indicator that p_k is Poisson distributed. The parameter for the Poisson distribution can be estimated as $\lambda = 5.2160$. Therefore, in the following, we assume that p_k follows a Poisson distribution and perform the analysis.

B. Effect of Selective Immunization

Our basic idea is to select an appropriate set of nodes to immunize. By immunization, we mean that these nodes are immune to the propagated mal-packets. In this way, the selected node can help disintegrate the network, thus preventing or slowing down the large-scale propagation of mal-packets. In practice, after we identify this set of nodes, we can take various approaches to immunize.

Therefore, our question becomes how to choose the appropriate set of nodes in order to partition the WSN network into as many separate pieces as possible.

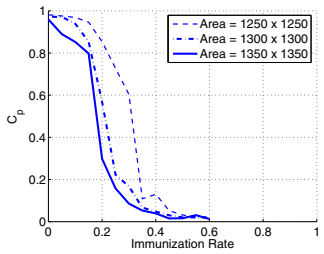
It is not a good idea to immunize all the sensor nodes because the potential large scale deployment of WSNs. Intuitively, the immunization of those most highly-connected nodes can help slow down the propagation of mal-packets because more connections between nodes are removed under this situation. In the following, we conduct a series of simulations to demonstrate the effect of selective immunization.

We simulate the deployment of $5,000$ nodes over square areas of $1,250 \times 1,250\ m^2$, $1,300 \times 1,300\ m^2$, and $1,350 \times 1,350\ m^2$, respectively. These deployments can lead to reasonable network densities. A sparse deployment which leads to a disconnected network can prevent the large-scale infection of mal-packets. On the other hand, if a network is densely deployed, for example, every sensor node has a connection to almost all the other nodes, it is helpless to immunize only a portion of nodes.

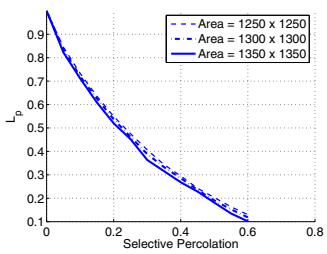
We use the following metrics to measure the effect of immunization:

- *Connection ratio* C_p: The ratio of the size of the largest component to the size of the remaining network when we remove the top p percent most connected nodes (and their related edges);

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

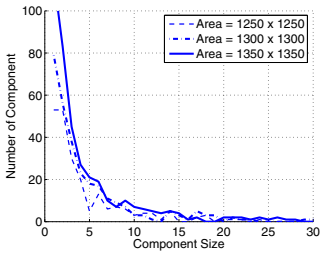


(a) Connection Ratio.

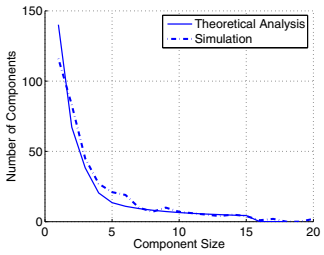


(b) Remaining Link Ratio.

Fig. 6. Simulation Results.



(a) Component Size Distribution after Immunization.



(b) Theory and Simulation - Component Size Distribution after Immunization.

Fig. 7. Results after Immunization.

- *Remaining link ratio* L_p: The fraction of remained links after we remove the top p percent of most-connected nodes;
- *Distribution of component sizes*: The distribution of the sizes of connected components in terms of nodes. Here, a connected *component* is a subset of vertices in the graph each of which are reachable from each other through some path.

In case the initial infected node starts infection from within the largest remaining component, the size of the largest remaining component gives the best situation that the infection can spread. Therefore, the distribution of component sizes can give us some idea about the infection effect.

1) Connection Ratio: Simulation results of the connection ratio are illustrated in Fig. 6(a). We have the following observations.

First, given a fixed immunization rate, the sparser the network, the smaller the largest remaining component becomes. This is what we expect.

Second, there exists a percolation threshold [12] for these simulations, where C_p drops dramatically when the immunization rate exceeds the threshold. For example, for the $1,350 \times 1,350\ m^2$ area, the threshold is roughly 0.22. We can see that the sparser the network, the smaller the percolation threshold becomes. This indicates that fewer nodes are needed to be immunized in order to disconnect the network.

We also observe a dramatic decrease of the largest component when the immunization rate reaches some value. For example, given the $1,300 \times 1,300\ m^2$ area, when the immunization rate reaches 30%, its largest component size drops dramatically. This phenomenon, which has been extensively studied in random graphs, is called percolation [12]. Therefore, in Section IV-B3, we further extend our study when 30% percent nodes are removed.

2) Remaining Link Ratio: Simulation results of the remaining link ratio are illustrated in Fig. 6(b). We observe a slight decrease of the remaining link ratio when the network becomes sparser. This is also what we expect.

3) Distribution of Component Sizes: Simulation results of the distribution of component sizes are illustrated in Fig. 7(a). We can see that when the network is sparser, after the removal of top 30% percent most connected nodes, there are more small components and fewer large components. Smaller components can slow down or prevent the propagation of mal-packets.

C. Analysis of Distribution of Component Sizes

The distribution of component sizes is an important indication of mal-packet propagation. In this section, with the help of random graph theory [12], we present the theoretical analysis of the distribution of component sizes.

First, we need to study the probability distribution of the number of second neighbors of one node, say node A, in one graph. To do this, we use q_k to denote the normalized distribution of the number of edges k emanating from vertex B other than the edge AB. Based on [12], we have

$$q_k = \frac{(k+1)p_{k+1}}{\sum_j j p_j} \tag{1}$$

where we recall p_j gives the probability that a randomly picked node has k neighbors. Here $\sum_j j p_j$ is the average degree of a vertex and its purpose is for normalization [12].

We further define the *probability generating function* for p_k as $G_0(x) = \sum_{k=0}^{\infty} p_k x^k$. The generating function for q_k is defined as $G_1(x) = \frac{G_0'(x)}{z}$ [12], where z denotes the mean number of neighbors of a randomly chosen vertex.

Randomly pick one edge in a graph, for example, edge AB. Following edge AB, we can reach vertex B. Consider the distribution of the sizes of those clusters reachable by node B and let $H_1(x)$ be the probability generating function that generates the distribution of the sizes of these clusters. Based on [12], we have

$$H_1(x) = x \sum_{k=0}^{\infty} q_k [H_1(x)]^k = x G_1(H_1(x)) \tag{2}$$

If we randomly pick one vertex, the distribution of the sizes of the clusters to which this randomly chosen vertex belong to is:

$$H_0(x) = x \sum_{k=0}^{\infty} p_k [H_1(x)]^k = x G_0(H_1(x)) \tag{3}$$

Assuming that p_k follows a Poisson degree distribution, as we have illustrated in Section IV-A, following the basic ideas proposed in [12], we have $G_0(x) = G_1(x) = e^{z(x-1)}$.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

Based on Equation (2), we have:

$$H_1(x) = x(q_0 + q_1 H_1(x) + q_2 {H_1}^{(2)}(x) + \ldots) \qquad (4)$$

Note that $q_0 = \frac{p_1}{<k>} = \frac{z^1 e^{-z}}{1!}/z = e^z$. Starting from $H_1(x) = q_0 x$, substituting this into the right part of Equation (4), and ignoring the part at order x^2 and higher, we have $H_1(x) = x(q_0 + q_1 q_0 x)$, where $q_0 = e^z$. Based on Equation (1), $q_1 = \frac{2p_2}{z} = ze^{-z}$. Therefore, $H_1(x) = xe^{-z} + (xe^{-z})^2 z$. Substituting this into the right part of Equation (3), we have $H_0(x) = x(p_0 + p_1(xe^{-z} + (xe^{-z})^2 z) + \ldots)$.

By computing the coefficient of $\frac{dH_0(x)}{dx^2}$, we can get the probability P_2 of a randomly chosen vertex belonging to components of size 2 is $P_2 = ze^{-2z}$.

Following this approach, we can iteratively compute the component size distribution for a given WSN network [12].

We simulate the empirical distribution of the component sizes. The simulation configuration that we use is $1,350$ by $1,350$ with $5,000$ nodes, in which top 35% percent most connected nodes are immunized. We use the Depth-First-Search algorithms to compute the component sizes. Based on our analysis in Section IV-A, we use a Poisson distribution to generate p_k and use the above methodology to calculate the theoretical distribution of component sizes.

The result is illustrated in Fig. 7(b). The curve corresponding to theoretical analysis presents the analytical results by the above methodology. Note that the theoretical analysis proposed in [12] is used for a very large network. Here we use a limited number of nodes which is typical for WSNs. This may account for the mismatch.

With the above analysis, we present a simple approach to identify immunization nodes in order to prevent or slow down the large-scale mal-packet propagation in WSNs. Based on how many nodes are deployed over how large an area, we can estimate the percolation threshold to disintegrate the network. This threshold enables us to calculate the *number of neighbor* threshold, denoted as N_{th}, above which one node should be immunized. Each node is then pre-equipped with N_{th}. After the deployment, each sensor node can measure connectivity information through a local $Hello$ protocol and count the number of one-hop neighbors, denoted as N_n. If N_n of one node A is larger than N_{th}, node A should be immunized. Node A can then report its location to the field officer.

After collecting these locations, some further actions can be taken. For example, different types of sensor nodes can be installed at these locations to avoid monoculture of the underlying WSN hardware. For a WSN consisting of Mica motes from *xbow* corporation, the more powerful IMote2 motes can be deployed at these locations. As a different example, for applications which are not time-sensitive, we can make sensor nodes deployed at these locations control and intentionally delay traffic flows in WSNs. In this way, the speed of mal-packet propagation can be effectively reduced.

V. Related Work

Few research works are focused on mal-packets on wireless networks. Yan *et al.* [7], [8] analyze the worm propagation in Bluetooth networks and investigate the impact of mobility patterns on Bluetooth worm propagation. Khayam *et al.* [9] propose a topologically-aware worm propagation model (TWPM) for WSNs. By incorporating MAC and network layer considerations, TWPM captures both time and space propagation dynamics. De *et al.* [10] model the node compromise in WSNs based on epidemic theory. Gu *et al.* [14] present attack approaches to construct specially crafted data message to facilitate mal-packet propagation in wireless sensor networks. Based on percolation theory, Zou [16] also apply immunization based approach to protect worm propagation in Email networks.

VI. Conclusions and Future Work

In this paper, we present a mal-packet propagation model in WSN and a preliminary analysis of this model using TOSSIM. Based on percolation and random graph theory, we propose an immunization based countermeasure to protect WSN. We select the most connected nodes to immunize and study their impact on preventing and slowing down mal-packet propagation. Future work includes demonstrating the effect of different network topology, the realistic characteristic of wireless sensor network including radio irregularity and transmission unreliability, and comprehensive evaluation of the proposed schemes.

References

[1] H. Kim and H. Cha, "Towards a Resilient Operating System for Wireless Sensor Networks," *2006 USENIX Annual Technical Conference*, Boston, MA, June 2006, pp. 103-108.

[2] http://www.f-secure.com/v-descs/cabir.shtml.

[3] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the Slammer Worm," *IEEE Security and Privacy*, Vol. 1, No. 4, 2003, pp. 33-39.

[4] C.-L. Fok, G.-C. Roman, C. Lu. "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications," *IEEE ICDCS'05*, Columbus, Ohio, June 6-10, 2005, pp. 653-662.

[5] K. Srinivasan *et al.*, "Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks," *ACM Sensys'06*, Boulder, CO, 2006, pp. 419-420.

[6] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in Your Spare Time," *11th USENIX Security Symposium (Security 02)*, Aug. 2002.

[7] G. Yan and S. Eidenbenz, "Bluetooth Worm: Models, Dynamics, and Defense Implications," *ACSAC'06*, Miami, FL, Dec. 2006, pp. 245-256.

[8] G. Yan and S. Eidenbenz, "Modeling Propagation Dynamics of Bluetooth Worms," *IEEE ICDCS'07*, Toronto, Canada, June 2007.

[9] S.A. Khayam and H. Radha, "A Topologically-Aware Worm Propagation Model for Wireless Sensor Networks," *IEEE ICDCS Int. Workshop on Security in Distributed Computing Systems*, June 2005, pp. 210-216.

[10] P. De, Y. Liu, and S.K. Das, "Modeling Node Compromise Spread in Wireless Sensor Networks Using Epidemic Theory," International Workshop on Wireless Mobile Multimedia, 2006, pp. 237-243.

[11] T. S. Rappaport, "Wireless Communications, Principles and Practice," *Prentice Hall*, 1996.

[12] M.E.J. Newman,"Random Graphs as Models of Networks," *Handbook of Graphs and Networks,* S. Bornholdt and H. G. Schuster (eds.), Wiley-VCH, Berlin 2003.

[13] B. Sun, G. Yan, and Y. Xiao, "Worm Propagation Dynamics in Wireless Sensor Networks," *IEEE ICC 2008*.

[14] Q. Gu, and R. Noorani, "Towards Self-propagate Mal-packets in Sensor Networks ," *ACM Wisec*, Alexandria, VA, 2008.

[15] P. Levis *et al.* "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *Sensys 2003*, Los Angeles, CA, 2003, pp. 126-137.

[16] C. C. Zou, Don Towsley, and W. Gong. "Modeling and Simulation Study of the Propagation and Defense of Internet Email Worm," *IEEE Tran. on Dependable and Secure Computing*, 4(2), 2007, pp. 105-118.