2008

Attack DistributionModeling and Its Applications in Sensor Network Security

Xiangqian Chen
Telecommunications and Information Technology Institute, Florida International University, xchen002@fiu.edu

Kia Makki
Telecommunications and Information Technology Institute, Florida International University

Kang Yen
Telecommunications and Information Technology Institute, Florida International University, yenk@fiu.edu

Niki Pissinou
Telecommunications and Information Technology Institute, Florida International University, pissinou@fiu.edu

Recommended Citation

X. Chen, K. Makki, K. Yen, and N. Pissinou, Attack Distribution Modeling and Its Applications in Sensor Network Security, Journal on Wireless Communications and Networking, 10.1155/2008/754252, 2008.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2008, Article ID 754252, 11 pages
doi:10.1155/2008/754252

Research Article

Attack Distribution Modeling and Its Applications in Sensor Network Security

Xiangqian Chen, Kia Makki, Kang Yen, and Niki Pissinou

Telecommunications and Information Technology Institute (IT2), Florida International University, Miami, FL 33174, USA

Correspondence should be addressed to Xiangqian Chen, xchen002@fiu.edu

Received 31 August 2007; Accepted 17 December 2007

Recommended by Farid Ahmed

Defending against attack is the key successful factor for sensor network security. There are many approaches that can be used to detect and defend against attacks, yet few are focused on modeling attack distribution. Knowing the distribution models of attacks can help system estimate the attack probability and thus defend against them effectively and efficiently. In this paper, we use probability theory to develop a basic uniform model, a basic gradient model, an intelligent uniform model and an intelligent gradient model of attack distribution in order to adapt to different application environments. These models allow systems to estimate the attack probability of each node under a given position and time. Applying these models in system security designs can improve system security performance and decrease the overheads in nearly every security area. Based on these models, we describe a novel probability secure routing algorithm that is effective to defend against attacks whether they are detected or not. Besides this application, we also introduce some other applications, such as secure routing that can save systems available energy and resources while still providing enough security, detecting attack, and key management.

1. INTRODUCTION

Recent advances in electronic and computer technologies lead to widespread deployment of wireless sensor networks (WSNs) on the horizon. Different WSNs may consist of different types of sensors, such as seismic, low sampling rate, magnetic, thermal, visual, infrared, acoustic, and radar sensors, which can monitor temperature, humidity, vehicular movement, lightning condition, pressure, soil makeup, noise levels, and so on [1]. These various classes of sensors lead to WSNs wide-range applications, including military sensing and tracking, environment monitoring, patient monitoring and tracking, and smart environments [2].

Many sensor networks have mission-critical tasks, such as above military applications. Thus, the security issues in WSNs are kept in the foreground among research areas. Compared with other wireless networks, such as ad hoc wireless LAN and cellular networks, security in WSNs is more complicated due to the constrained capabilities of sensor node hardware and the properties of the deployment environment .

Security issues mainly come from attacks. If no attack occurred, there is no need for security. Thus, detecting and defending against attacks are important tasks of security mechanisms. It is obvious that knowing the probabilities of attacks can help systems monitor, identify, and defend against them efficiently and effectively. Although there are some approaches that can be adapted to detect and defend against attacks, few of them have been done to provide a method to estimate the probability of being attacked for each node. Most current approaches assume the same probability of attack occurring everywhere as a matter of course, and use this embedded assumption without a clear declaration in their systems. In fact, their hypothesis is different from some special applications in which attacks may occur with different probabilities. For example, how can one think that the attack close to an enemy-controlled area transpires with the same probability as in a controlled area?

In this paper, we present several attack distribution models in order to estimate attack probability, and then provide several applications based on these models. Our current modeling works are based on static WSNs, that is, sensor nodes will not change their positions after deployment. Besides this assumption, we suppose that there exists attack detecting system in our intelligent models. Our current attack distribution models can be adapted to those types of attacks that the attack probability for a node is correlated with the attack events of its neighbors and its position. In WSNs, many types of attacks occur with the above neighbor effect and position effect. Based on our survey, this is the first time that attack distribution models have been proposed to estimate the attack probability of a node under a given position and time. The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 describes the details of attack distribution models. Section 4 shows some applications of these models. Finally, we conclude and lay out some future work in Section 5.

2. RELATED WORK

In this section, we give a concise introduction of related work as two categories: attack detection and prevention, and node positioning.

2.1. *Attack detection and prevention*

Due to the wireless nature and special deployment environments of WSNs [3], a great variety of attacks are possible. To express clearly, we give a short summation of attacks and defense suggestions based on the point of view of open system interconnect (OSI) model. Generally, the typical layered networking model of sensor networks includes the physical layer, the data link layer, the network layer, the transport layer, the middleware layer, and the application layer. Each layer is susceptible to different attacks. Even some attacks can crosscut multiple layers or exploit interactions between them. In this paper, we mainly discuss attacks and defenses on the transport layer and below layers.

Physical layer

Jamming and tampering are the major types of physical attacks [4]. The standard defense against jamming involves various forms of spread spectrum, frequency hopping, low-duty cycle, rerouting traffic, adopting prioritized transmission scheme, and so on. Tampering is another type of physical attack in sensor network. An attacker can also tamper with nodes physically, interrogate and compromise them. Tamper protection falls into two categories: passive (e.g., hiding) and active (e.g., tamper-proofing circuit).

Data link layer

Collision, exhaustion, and unfairness are the major attacks in this layer [4]. The normal defending methods to these three attacks, respectively, are error-correcting code, rate limitation, and small frames, although these mechanisms have limitations.

Network layer

There are many types of attacks in this layer. Karlof and Wagner summarize the attacks of network layer as follows: spoofed, altered, or replayed routing information; selective forwarding; sinkhole attacks; sybil attacks; wormholes; HELLO flood attacks; acknowledgement spoofing [5]. Authentication, identification, multipath, neighbor node monitor, location, distance verification, and so on are the normal methods to prevent routing attacks.

Transport layer

Flooding and desynchronization are the normal attacks in this layer [4]. Solving client puzzles can partly ease flooding. One counter to desynchronization is to authenticate all packets exchanged, including all control fields in the transport protocol header.

As a whole, attack detecting methods can be classified as centralized approaches and neighbors' cooperative approaches. Centralized approaches use the base station to detect attacks [6, 7]. In neighbors' cooperative approaches, neighbor nodes of the given node collect neighbors' information and make a collective decision to detect attacks [8, 9]. Essentially, [10] is a neighbors' approach because it collects neighbors' data, though it processes them with statistical method. Similarly, [11] also belongs to neighbors' approach, though it makes decisions based on threshold analysis.

We note that all of the above schemes can be used to detect attacks in some extent; however there might not be high efficiency because researchers implicitly suppose that any node, whether it is located near or far from the base station, has the same probability of being attacked. This assumption is not always suitable; for example, in battlefield surveillance applications, the attack event close to an enemy-controlled area occurs with a larger probability than in a controlled area. Thus, knowing the distribution of attacks can help us to design efficient and effective secure mechanisms to detect and defend against them. This point is our main focus in this paper.

2.2. *Node positioning*

In some location systems, several sensors have a position system such as GPS to locate their positions. We call this type of sensor beacon node. These location systems use location information from these beacon nodes to construct the whole location system by utilizing ultrasound and time-of-flight techniques. Capkun and Hubaux [12] proposed a mechanism for position verification, called verifiable multilateration (VM) based on distance bounding techniques [13], which can prevent a compromised node from reducing the measured distance. VM uses the distance bound measurements from three or more reference points (verifiers) to verify the position of the claimant. Lazos and Poovendran [14] proposed a range overlapping method instead of using expensive distance estimation methods. Its main idea is as follows: each locator transmits different beacons with individual coordinates and coverage sector areas. After receiving enough sector information from different locators, the sensor estimates its location as the center of gravity of the overlapping region of the sectors that include it.

Due to adversaries' attacks, the beacon nodes or normal nodes maybe compromised. Some location systems estimate location by combining deployment knowledge and probability theory without beacons. For example, Fang et al. [15] integrated predeployment knowledge of sensors and the maximum likelihood estimation method to estimate the sensors' locations.

3. MODELING OF ATTACK DISTRIBUTION

Before presenting the models of attack distribution, we describe some assumptions regarding the sensor network security scenarios.

3.1. Network and security assumptions

The followings are assumptions of WSNs.

(i) Base station: the base station is computationally robust, having the requisite processor speed, memory, and power to support the cryptographic and routing requirements of the sensor network. Adversaries can destroy the base station but they cannot compromise it within the limited time.

(ii) Sensor nodes: the sensor nodes are similar to current generation sensor nodes in their computational and communication capabilities and their power resources [16]. They can be deployed via aerial scattering or by physical installation. We assume that any sensor node will know the position of itself and its immediate neighbor nodes after deployment and the base station will know all the nodes' positions. All the sensor nodes will not change their positions after deployed. If adversaries change the positions of nodes or identity, the neighbor nodes will detect this attack [17], and this is not the focus of this paper.

(iii) Adversary: adversaries have unlimited energy and computing power. An attacker needs to spend some time to attack a node. In the attacking process, they will not change the targets until the chosen target nodes were attacked. After attacking one node, the attacker will continue attacking a new good node without any halt, stop, or hibernation.

3.2. Distribution models

Based on whether, an attack event is thought of as independent event or not; we classify the attack distribution models as either basic models or intelligent models. To focus on the main viewpoint of attack distribution models, we only use 2-dimension distribution models, which assume that all the nodes are in the same plane.

3.2.1. Basic attack distribution models

We label some models as basic attack models because the probability of one sensor being attacked does not affect its neighbors within these models. When the attack probability and the frequency are comparatively small, the correlation of attacking among neighbors can be neglected. Under this condition, basic models are accurate enough to estimate the attack probability. Due to different application environments, we classify the basic models as either uniform models or gradient models.

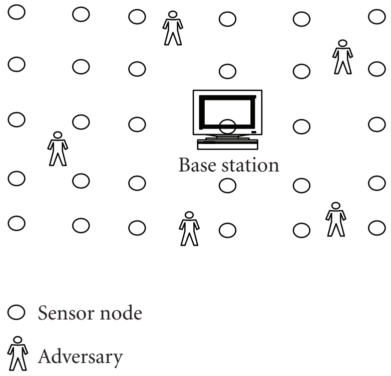

Figure 1: Basic uniform attack model.

(1) Basic uniform attack distribution model

In some sensor network application situations, such as environmental and health applications, every sensor node has nearly the same probability of being attacked despite of its position. In such cases, the attack probabilities of nodes following uniform distribution are reasonable, as shown in Figure 1.

The mathematical model is given by

$$P_{(x,y,t)} = \rho(t), \tag{1}$$

where (x, y) is the coordinate of the sensor; $p_{(x,y,t)}$ is the attack probability of this sensor at time t; $\rho(t)$ is a distributed function which is independent of the coordinates of a sensor. Most current security approaches use this simple model without a clear declaration.

(2) Basic gradient attack distribution model

In some special application scenarios, such as battlefield surveillance, reconnaissance of opposing forces and terrain, and other military applications, the basic uniform attack model is not suitable because the nodes close to an enemy-controlled area may have larger probabilities of being attacked than the nodes that are far away from an enemy-controlled area. Thus, a rough gradient-based attack model approximates to the real environment. The gradient is based on the distance from the opponent or the base station, as shown Figure 2.

The mathematical model is given by

$$P_{(x,y,t)} = \rho(0,0,t)\left(1 + gd_{(x,y)}\right), \tag{2}$$

where $\rho(0,0,t)$ is the attack probability in the base station area at time t; g is the gradient function; $d_{(x,y)}$ is the projective vector of sensor (x, y) in the gradient direction. In

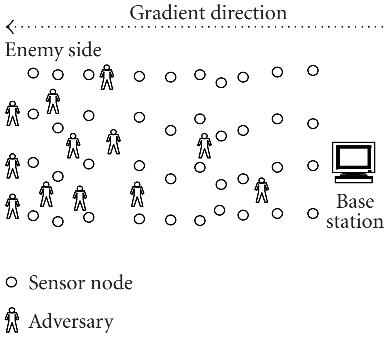

Figure 2: Basic gradient attack model.

this model, the closer that a sensor node is to an enemy-controlled area, the more probable that it is attacked. *The difference between a uniform model and a gradient model is that the location of a sensor may affect the attack probability in the latter model, while it does not matter in the previous model.*

3.2.2. Intelligent attack distribution models

The above basic models assume that every attack is an independent event. This supposition is not accurate enough when the probability and frequency of attacks are comparatively larger, especially in a dense sensor network. In this environment, the attack probability will increase when its neighbors have been recently attacked. It is easier and more conceivable for adversaries to attack the nearest neighbors in the next period after they have attacked a sensor because of what follows.

(i) The communication information between the attacked node and its neighbors may help adversaries to attack them easily, and the adversary is intelligent enough to utilize this correlation.

(ii) A recently attacked node means that the adversary is close to that node, and thus its neighbor nodes have larger probabilities of being chosen as the target of this adversary.

(iii) Attacking more nodes in a nearby area may badly impair the system when the sensor network uses a majority decision mechanism to integrate data, prevent error, and so on.

The difference between a basic model and an intelligent model is that the latter model considers the effect of attack events coming from neighbor nodes when estimating the attack probability. In intelligent models, systems should have mechanisms to detect and record the attack events and use current attack events to estimate future attacks. That's why we call these models intelligent models. Before describing intelligent models, we give some technical terms as follows.

(i) Attacked node: it is a node that has already been attacked by an attacker and the attacker got its assaulting result, such as compromising the node, disabling it, and so on.

(ii) Attacking time: the time spent by an attacker to attack a benign node to get assaulting result. In our models, attacking time follows normal distribution and the expected value is τ.

(iii) Detected attacked node: it is an attacked node and the attack event has already been detected by the system.

(iv) Recently attacked node: it is an attacked node that has been attacked within time interval τ.

(v) Detecting attacked time: the time interval between the time when the node was attacked and the time when the system detected that the node was attacked. In our models, it also follows normal distribution.

(vi) i-hop neighbor: an i-hop neighbor is a node that at least needs number of i-hops to reach the given node.

In this type of model, we assume that the expected time for an adversary attack against a good node is τ and adversaries will continue attacking the good nodes with this frequency without any halt, stop, changing attacking target, or hibernation. In some sensor security mechanisms, the expected value τ maybe decreases when more and more nodes are attacked. But the attack difficulty can be retained as the previous and the assumption of the average attack time is still suitable if the application meets one or two cases: the total number of the attacked nodes is comparatively small compared with the large number of the normal nodes; the system assumes some adapting methods to enhance the security. A normal distribution with expected value τ can approximate the attack probability. Under this assumption, we time the system with each interval of τ. Our object is to use current available attack event information to estimate the attack probability in the next time period. We imagine that the probability of a node being attacked includes two parts: current adversaries and new adversaries, which will be joined in the next period. Thus, we get the following mathematical model:

$$P_{(x,y,t)} = S_{(x,y,t)} + C_{(x,y,t)} \quad (n\tau < t < (n+1)\tau,\ n = 0,1,2,\ldots), \tag{3}$$

where $S_{(x,y,t)}$ is the attack probability, which is introduced by newly added adversaries in the time period from $n\tau$ to $(n+1)\tau$; $C_{(x,y,t)}$ is the probability that is introduced by current adversaries.

Similar to basic model classifications, an intelligent model can also be classified as a uniform model and a gradient model.

(1) Intelligent uniform attack distribution model

This model adapts the application environment where the new adversaries evenly distribute within the coverage area. In this model, (3) can be expressed as follows:

$$P_{(x,y,t)} = S_{(t)} + C_{(x,y,t)} \quad (n\tau < t < (n+1)\tau,\ n = 0,1,2,\ldots), \tag{4}$$

where $S_{(t)}$ follows uniform distribution and does not care about node positioning, and this part is introduced by newly added adversaries from time $n\tau$.

We assume 1-hop neighbors of the given node are the nodes which are the immediate neighbor nodes of the given

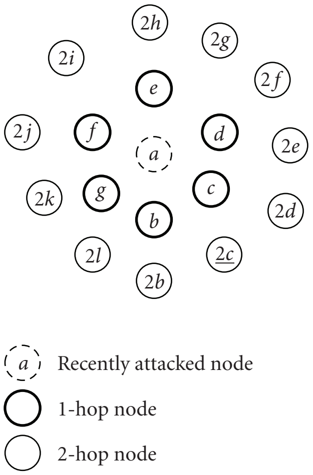

Figure 3: Difinitions in intelligent mode.

node and can directly connect to this node; 2-hop neighbors of the given node are the nodes which can contact the given node at least by two hops, and so on. We call all the 1-hop neighbors of the given node as 1-hop layer nodes, and all the 2-hop neighbors as 2-hop layer, and so on. In dense WSNs, the distances between a given node and its 1-hop neighbors are nearly equal. Therefore, we suppose that each 1-hop benign neighbor of a recently attacked node has the same probability of being chosen as the attacking target of an adversary which corresponds to this recently attacked node. Similarly, we make the same assumption of 2-hop neighbors, 3-hop neighbors, and so on. While the probability that one of 1-hop layer nodes being chosen as the attacking target is larger than the probability of 2-hop layer node, and so on, a geometric distribution can approximate the probability of the adversary, which corresponds to the recently attacked node, choosing an attacking target from different layers.

Figure 3 clearly shows the above definitions. As shown in Figure 3, node a is the given node; nodes b, c, d, e, f, and g are 1-hop neighbor nodes of node a; nodes $2b - 2l$ are 2-hop neighbors of node a. Nodes b and g have the same probability of being chosen as the attacking target in the next time period. Similarly, nodes $2b$ and $2l$ have the same probability of being chosen as the attacking target in the next time period. While the probability that one of 1-hop layer nodes (b and g) being chosen as the attacking target is larger than that of one of 2-hop layer nodes ($2b$ and $2l$), and so on, a geometric distribution can approximate this assumption.

As shown in Appendix A, $C_{(x,y,t)}$is given by

$$C_{(x,y,t)} = 1 - \prod_{i=1}^{N}\prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)\right) \qquad (5)$$
$$(n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots),$$

where N is the largest number of hops that node (x, y) can access all the nodes in the network; M_i is the number of nodes that have been recently attacked and are i-hops to node (x, y); node (x_{ij}, y_{ij}) is denoted as the jth recently attacked node in all M_i nodes; n_{ij} is the total number of i-hop neighbors to node (x_{ij}, y_{ij}) and k_{ij} of them are attacked nodes; the probability of one of i-hop nodes to be chosen as the attacking target of the adversary, which corresponds to a recently attacked node, is p_i. $Q_{ij}(t)$ is the attack probability of the chosen attacking target in time t. $Q_{ij}(t)$ follows normal distribution and the expected value is τ. p_i follows geometric distribution and is given by

$$p_i = ar^{d(i-1)} \qquad (i = 1, 2, \ldots,\ 0 < a < 1,\ 0 < r < 1), \qquad (6)$$

$$\sum_{i=1}^{N} p_i = 1, \qquad (7)$$

where a, r, and d are parameters of geometric distribution; a is the total probability of an adversary choosing a good node, 1-hop to the recently attacked node, as the attacking target; r is the ratio which is less than 1, and d is a natural number.

As shown in Appendix A, we get the following equation:

$$a = 1 - r^d. \qquad (8)$$

In the case of $n_{ij} = k_{ij}$ in (5), we use 1 instead of the product item $1 - 1/(n_{ij} - k_{ij})p_i Q_{ij}(t)$ first, and then replace p_{b+1} with $p_b (b > i)$ for each product item with index j. For example, if $n_{1j} = k_{1j}$, we use 1 instead of the product $1 - 1/(n_{1j} - k_{1j})p_1 Q_{1j}(t)$, and replace p_2 with p_1, p_3 with p_2, and so on for each product item with index j.

In normal distribution, about 99.7% of values lie within 3 standard deviations. The beginning attacking time (denoted by t_s) is the time when node (x_{ij}, y_{ij}) is actually attacked. In time $t_s, Q_{ij}(t)$ is equal to 0. In a practical environment, we cannot know the actual attacking time $S_{(t)}$, but we can approximate it by subtracting the average detecting time from the actual detecting time of node (x_{ij}, y_{ij}) being attacked.

Suppose the number of new added adversaries follows uniform distribution of time. As shown in Appendix B, S_t is given by

$$S_{(t)} = 1 - \prod_{i=0}^{m-1}\left(1 - \frac{\lambda \Delta t}{N_g} Q(n\tau + i\Delta t)\right) \qquad (9)$$
$$(n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots),$$

where Δt is a very small time period which can be thought of as the smallest time unit in the system; $t = n\tau + m\Delta t$ $(m = 1, 2, 3, \ldots)$; λ is the number of new adversaries that are introduced in a unit time; N_g is the number of current good nodes in the network; Similar to $Q_{\ddot{u}}(t)$, $Q(n\tau + i\Delta t)$, follows the same normal distribution and $n\tau + i\Delta t$ is the time when newly introduced attacker nodes begin to attack probability in a unit time for each node (i.e., a node has δ probability of being chosen as the attacking target by the new adversaries in a unit time), which is given by

$$\delta = \frac{\lambda}{N_g}. \qquad (10)$$

To describe clearly the intelligent uniform model, we use Figure 4 to calculate the attack probability of node a.

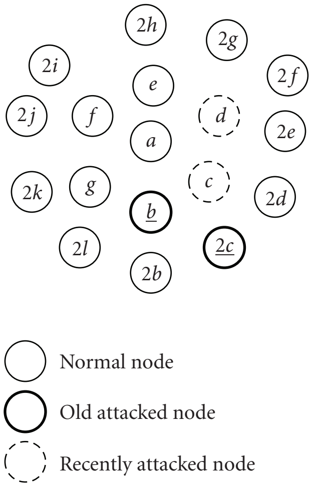

Figure 4: Intelligent uniform attack mode.

In Figure 4, nodes b and g are 1-hop neighbors of node a; nodes $2b$ and $2l$ are 2-hop neighbors of node a; nodes $a, c, 2c, 2b, 2l$, and g are 1-hop neighbors of node b; nodes b and $2c$ are recently attacked nodes that have been attacked in the last time period; nodes d and c are old attacked nodes. In Figure 4 for node a, $N = 2$, that is, node a can reach all the sensors in the network within 2 hops. Node a has one 1-hop neighbor node (node b) and one 2-hop neighbor node (node $2c$) that have been recently attacked. So $M_1 = 1$ and $M_2 = 1$. Node b has six 1-hop neighbors, thus $n_{11} = 6$. Node b has two 1-hop attacked neighbors, that is, node c and node $2c$, then $k_{11} = 2$. Node $2c$ has five 2-hop neighbors (node $2l, g, a, d$, and $2e$) and one 2-hop attacked neighbor (node d), consequently $n_{21} = 5$, $k_{21} = 1$. Suppose $p_1 = 0.8, p_2 = 0.16, Q_b(t) = 0.6, Q_{2c}(t) = 0.4$, and no new adversaries are introduced in the network. We calculate the attack probability of node a as follows:

$$\begin{aligned} P_{(x,y,t)} &= 0 + \left[1 - \left(1 - \frac{1}{6-2} \times 0.8 \times 0.6\right)\right. \\ &\qquad \left.\times \left(1 - \frac{1}{5-1} \times 0.16 \times 0.4\right)\right] \\ &= 0.13408. \end{aligned} \tag{11}$$

(2) Intelligent gradient attack distribution model

This model adapts the application environment where the new introduced attackers follow a gradient distribution of positions. Similar to the above intelligent uniform model, the mathematical model of attack probability is given by

$$\begin{gathered} P_{(x,y,t)} = S_{(x,y,t)} + \left[1 - \prod_{i=1}^{N}\prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)\right)\right] \\ (n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots), \end{gathered} \tag{12}$$

where $S_{(x,y,t)}$ is given by

$$\begin{gathered} S_{(x,y,t)} = \rho(0,0,t)\left(1 + g d_{(x,y)}\right) \\ (n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots). \end{gathered} \tag{13}$$

Equation (13) is similar to (2). The only difference between these two equations is that the intelligent models partition the system time in small time period, which equals the average attacking time τ. The only difference between an intelligent uniform model and an intelligent gradient model is that they have different first items in the mathematical model expression. The first item of the latter follows a gradient distribution of position, while the previous follows a uniform distribution. Similar to an intelligent uniform model, $\rho(0,0,t)$ can be estimated as the following equation:

$$\begin{gathered} \rho(0,0,t) = 1 - \prod_{i=0}^{m-1}\left(1 - \delta_0 \Delta t Q(t_0 + i\Delta t)\right) \\ (n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots), \end{gathered} \tag{14}$$

where δ_0 is the attack probability in a unit time in the base station area (i.e., a node has δ_0 probability of being chosen as the attacking target in a unit time in this small area); the other parameters in (14) are the same as parameters in (9).

Someone may say that the second part of (12) should also adjust with gradient weight. Firstly, for a given recently attacked node, the probability of a corresponding adversary choosing an 1-hop layer node as the attacking target is larger than the probability to choose a 2-hop layer node (i.e., $p_1 > p_2 > p_3 \ldots$). Secondly, the difference of gradient weight among 1-hop neighbors is comparatively small especially in dense networks. Thirdly, for an attacker, the difference of attacking probabilities in different directions is close to zero. The number of attackers in different directions can embody the gradient model enough. Thus, for easy estimation, we only introduce the gradient vector in the first part of (12). Figure 5 shows this model.

4. APPLICATIONS OF ATTACK DISTRIBUTION MODELS

Defending against attacks is the key successful factor for sensor network security. Attack distribution model can help systems defend against attacks before they occur or if they have already occurred but have not been detected. Our models can be applied to many types of attacks. For example, basic models can be adapted to most types of attacks that are introduced in Section 2. And they provide a rough attack probability estimation that can be used to analyze system security weakness and help to defend against them with more efficiency and effectiveness. While our intelligent models can be applied to detect and defend against those types of attacks that have neighbor correlation effects with giving more accurate attack probability estimation , a neighbor correlation effect is a phenomenon that a node has larger probability of being attacked in the near future when its neighbor has been recently attacked. Of course, to use intelligent models, systems have many attack detecting mechanisms.

We can apply attack distribution models to analyze system security weakness, improve security performance,

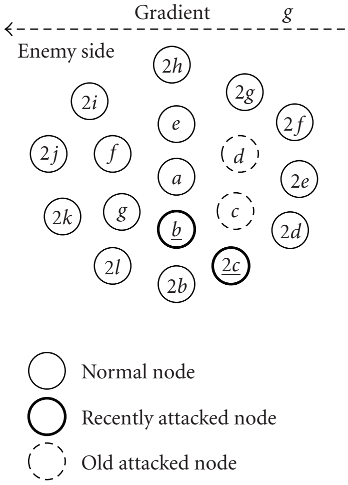

Figure 5: Intelligent gradient attack model.

distribute system resources efficiently on security cost, and so on. Because this is the first introduction of the attack distribution model, more research works should be performed in the future. In the following, we will give some application examples of how to use our models to provide efficient and effective security mechanisms.

4.1. Detecting attack

Detecting attack is an important task for system security. In this area, the modeling of attack will help a lot. A standard application of intelligent models is to integrate them into current attack detecting system. For example, most current monitoring systems, such as in [6–11], monitor all the nodes in the system without emphasis, and the system should decentralize their resources evenly in all nodes in order to monitor whether they have larger attack probabilities or not. That makes the detecting mechanism less efficient. Due to the heavy work, the system performance may decrease largely, and may even make this work unpractical. Applying our models to these monitoring systems and choosing nodes that have larger attack probabilities as the main monitoring objects will make node monitoring work more effectively and more efficiently; thus allowing the system to have enough resources to defend against attacks.

4.2. Secure routing

WSNs use multihop routing and wireless communication to transfer data, thus incur more routing attacks. To our knowledge, there is no previously published work to provide an effective routing algorithm that can prevent routing paths from passing those nodes that have been attacked but have not been detected by the system. Based on our survey, until now few proposals even consider undetected attack issues.

An ideal secure routing algorithm to defend against attacks lets routing paths bypass all attacked nodes. However, most attack activities can not be immediately detected because any detection mechanism needs time and the fraudulent action of adversaries (Adversaries do not want system to notice their attacking activities, thus they will adopt any action that one can imagine to make the detecting time longer.) makes the time even longer. A routing path is still a compromise path when it passes those “good” nodes which system considers as good nodes while they are actually attacked nodes that just have not been detected yet. Applying our attack distribution models in secure routing algorithm design can ease this issue. We develop a novel probability secure routing scheme that estimates the attack probability and makes the routing paths detour those nodes that have already been detected as attacked nodes or have larger attack probabilities than the given threshold.

Figures 6, 7, and8 are the results from the same simulation. These three figures are used to compare different routing algorithms. To describe easily, we define the routing algorithm without security consideration as ALG-I (e.g., AODV in [18]), the algorithm that the routing path bypasses those detected attacked nodes as ALG-II (e.g., pathrater in [8]), our algorithm as ALG-III (*threshold is* 0.12). The threshold choosing corresponds to the security requirement. We will discuss it later. In this simulation, the attack distribution follows an intelligent uniform model; there are 400 sensor nodes in the network and node density is equal to 10. The expected time for an adversary to attack a benign node is τ, which is equal to 300 unit time; the average time for system to detect an attacked node is also equal to τ. In each unit time, there are 10 randomly chosen routing requests to the base station; the simulation time is 20τ; the intelligent model parameters values are as follows: $a = 0.8, r = 0.2$, and $d = 1$. At the beginning time of this simulation, there are 10 adversaries introduced to attack this sensor network, and there are no more newly adversaries to be introduced in this system. The probability threshold to distinguish good or bad nodes is 0.12.

In Figure 6, average compromise path ratio is the ratio of the number of compromise paths to the number of routing requests in the whole simulation time. If the value of average compromise path ratio is larger, it means less routing security under attack. This figure clearly shows what follows: the average compromise path ratio in ALG-I is the largest among three algorithms; the average compromise path ratio in ALG-II is in the middle; ALG-III has the least average compromise path ratio as expected, and has the best security performance. That’s easy to understand. ALG-I has the largest probability of finding a routing path to pass attacked nodes because the routing algorithm does not consider detouring attacked nodes. The attack probability will be rapidly decreased when the system adopts attack detecting mechanisms and makes the routing paths bypass those attacked nodes that have been detected by the system. Besides bypassing those detected attacked nodes in the routing path, our algorithm also lets the routing path bypass those nodes that have larger probabilities of being attacked, and then the routing path will bypass some nodes that have already been attacked but have not been detected by the system. As a result, our algorithm improves the routing security further.

Figure 7 compares the average routing path length (it is the average number of links for each routing path.) in

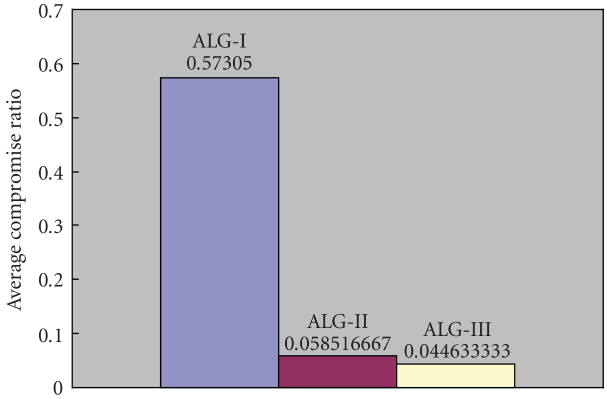

Figure 6: Routing security comparison.

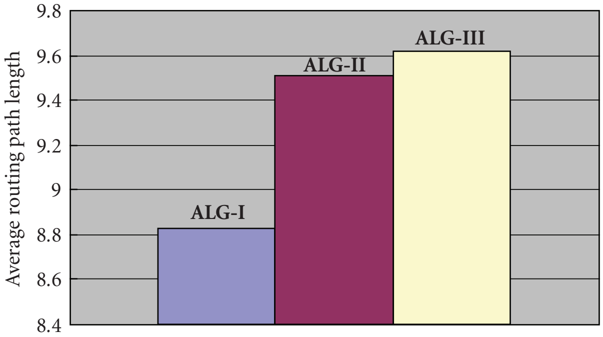

Figure 7: Routing overhead comparison.

different algorithms. It shows what follows: the average path length in ALG-I is the smallest; the average path length in ALG-II is in the middle; ALG-III has the largest average path length. The reason is that ALG-I finds the routing paths that have the least hops, thus it has the smallest average path length; while ALG-II may find paths that satisfies the security requirement but may not be the least hop paths. In our algorithm, besides bypassing those detected attacked nodes in the path, the routing path should also detour some estimate bad nodes, making the average path length the largest among the three types of algorithms.

Figure 8 compares the average successful ratio (This is the ratio of the number of successful routing requests to the total number of routing requests.) in different algorithms. It shows what follows: the average successful ratio is 100 percent in ALG-I; the average successful ratio in ALG-II is in the middle; the average successful ratio in ALG-III is the least. Radically, in a completely connected network, every routing request will find a successful path. While some routing requests cannot find successful routing paths in ALG-II because there exist some probabilities for some nodes who are surrounded by detected attacked nodes and cannot find valid routing paths; the successful ratio will decrease further when the system considers some probability attack nodes as bad nodes in our algorithm.

Figures 9, 10, and11 compare the security, overhead, and successful ratio results with different thresholds in our algorithm. We use the same parameters as the simulation for Figures 6, 7, and 8, except different thresholds.

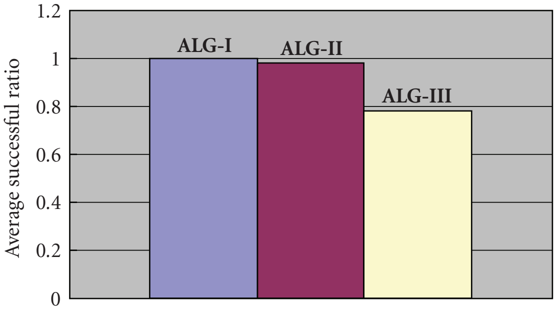

Figure 8: Successful routing ratio comparison.

The main object of Figures 9 and 11 is to compare security, overhead and successful routing effects under different thresholds. When the threshold increases, the security performance decreases (average compromise path ratio increases as shown in Figure 9), the average length of routing paths decreases (average path length decreases as shown in Figure 10), and successful ratio increases (average successful ratio increases as shown in Figure 11). The reason is that after the threshold increase, the system considers more nodes as good nodes and it makes the secure network connectivity increase. Thus, the system has a larger probability to find a successful routing path for a routing request, and the average length for routing paths decreases because the total number of bad nodes in the algorithm is getting smaller. At the same time, the security performance decreases because a routing path has a larger probability to pass a node that has actually been attacked but has not been detected, and is thought of as a good node in the system. These three figures also show that the curves change sharply initially and tend to flat later. The reason is that we suppose that attacking time follows a normal distribution, and the attacking time for most attack events will fall into the nearby area of the expected value of the normal distribution (normal distribution has a convergence property). If the threshold is close to the center value of the above converging area, then the number of undetected attacked nodes to be filtered by our algorithm will vary to a large extent, making the curves tilt sharply. While the threshold is far from the center value of the above converging area, the number of undetected attacked nodes to be filtered by our algorithm will alter less, making the slope of the curves a near constant.

Besides improving routing security, using our models can also help systems save effective energy. As we know, systems cannot use attacked nodes in some applications, though they may have larger energy. If we know a node has a larger probability of being attacked in the future, utilizing its resources and energy before it has been attacked will help systems decrease the energy and resource loss. Attack distribution models can estimate attack probabilities in the future. If we apply attack distribution models and design a routing algorithm which allows routing paths to choose those nodes whose attack probabilities are still in the secure scope but may enter

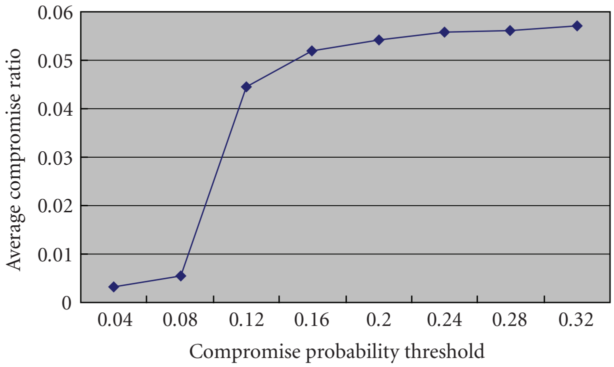

Figure 9: Routing securities in different thresholds.

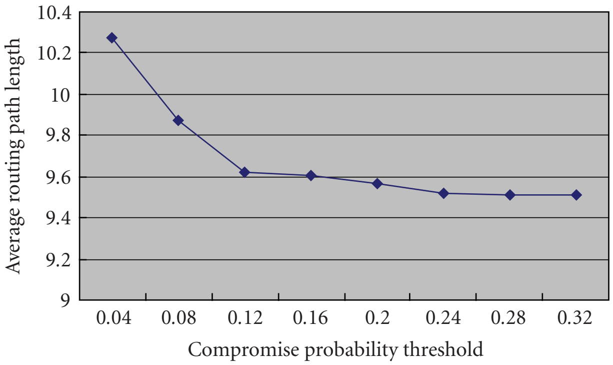

Figure 10: Routing overhead in different thresholds.

into an insecure scope in the future, it will save systems effective energy and resources while still providing enough security.

4.3. Key management

For security, key management is very important and complex, especially in symmetric cryptography structures. Many current key management proposals, such as [19–21], do not consider the attack distribution. They imply the attack probability to be the same for every node. However, when their security system is deployed in a different environment from their supposition, the security performance will decrease greatly.

For example, in [19], the security scheme requires q common keys (q is a constant, $q \geq 1$) to establish secure communications between a pair of nodes. In their scheme, q is equal in each area. When their scheme is deployed in a gradient-based environment, the security performance will decrease because the system has the same ability to tolerate or defend against attacks in all areas, but adversaries attack the system with different strengths on different areas; thus making the system unable to provide enough security in some areas and able to provide more security than needed in other areas. Of course, you can increase q to provide enough security everywhere, but it will consume more resources. It looks difficult to get a high security performance with a low overhead; however, when you apply an attack distribution model to this security mechanism, you will find that this is the key in solving this issue. For example, if we apply q to follow the same distribution as the attack distribution model, that is, $q \Rightarrow q(x, y)$, where (x, y) is the coordinates of node, the system will solve the above-mentioned issue. In the modified security scheme, the ratio between the strength of preventions and attacks can be kept the same in every area. In [20, 21], though this scheme has a nice threshold property λ (when the number of compromised nodes is less than the threshold λ, the probability that any nodes other than these compromised nodes are affected is close to zero), it needs more resources to implement this desirable threshold when it is deployed in a gradient-based application environment. Similarly, we can also apply λ to follow the same distribution as the attack model of the given application environment to ease the issue.

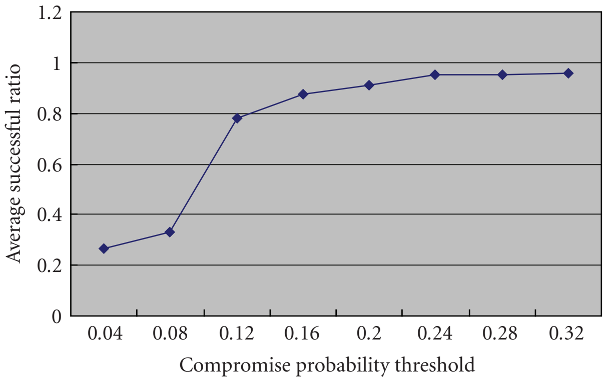

Figure 11: Successful routing ratio in different thresholds.

Besides improving the key predistribution step of key management, we can also apply our models to aberrant node management, rekeying frequency, and so on. with the similar modification method in order to improve system performance and security.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed several models to estimate attack distribution in different sensor network application environments. These models allow systems to estimate the probabilities of attacks. Applying these models to system security design will improve system security performance and decrease the overheads in nearly every security related area. Based on these models, we briefly describe a novel secure routing algorithm that can defend against undetected attacks effectively. Besides this application, we also introduce some other applications, such as secure routing that both saves systems available energy and resources while still providing enough security, detecting attack, and key management.

Because this is the first time we try to model the distribution of attacks, there are some important works that we plan to study in the future. For example, how to model the attack distribution in mobile networks? How to find the suitable values for the parameters in current models when they are deployed in practical applications?

APPENDICES

A. APPENDIX A

In the intelligent uniform model, the probability of a node being attacked which is introduced by all recently attacked nodes is given by

$$C_{(x,y,t)} = 1 - \prod_{i=1}^{N}\prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)\right) \quad (n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots,). \tag{A.1}$$

Suppose

Benign node (x, y) can access all the nodes in the network at most by N hops; node (x, y) has M_i recently attacked nodes which are i-hops to it. We denote node (x_{ij}, y_{ij}) as the jth recently attacked node in all M_i nodes; node (x_{ij}, y_{ij}) has n_{ij} i-hop neighbors and k_{ij} of them are attacked nodes; the probability of one of i-hop nodes of being chosen as the attacking target of the adversary, which corresponds to a recently attacked node, is p_i. p_i follows geometric distribution and is given by

$$p_i = ar^{d(i-1)} \quad (i = 1, 2, \ldots,\ 0 < a < 1,\ 0 < r < 1), \tag{A.2}$$

$$\sum_{i=1}^{N} p_i = 1, \tag{A.3}$$

where a, r, and d are parameters of geometric distribution, d is a natural number.

From (A.3), we have following equation:

$$\sum_{i=1}^{N} p_i = a + ar^d + ar^{2d} + ar^{3d} + \cdots = \frac{a - ar^{dN}}{1 - r^d}. \tag{A.4}$$

If N is a large natural number, (A.4) can be expressed as the following equation:

$$\sum_{i=1}^{N} p_i \approx \frac{a}{1 - r^d}. \tag{A.5}$$

From (A.3) and (A.5), we get the following equation:

$$a = 1 - r^d. \tag{A.6}$$

Derivation of $C_{(x,y,t)}$

From the above suppositions, the probability (denoted by e) of node (x, y) to be chosen as the attacking target of the adversary which corresponds to node (x_{ij}, y_{ij}) is given by

$$e = \frac{1}{n_{ij} - k_{ij}} p_i. \tag{A.7}$$

The probability (denoted by f) of node (x, y) of being attacked at time t, which corresponds to node (x_{ij}, y_{ij}), is given by

$$f = \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t), \tag{A.8}$$

where $Q_{ij}(t)$ is the attack probability of the chosen attacking target in time t. $Q_{ij}(t)$ follows normal distribution and the expected value is τ. Thus, the unattacked probability (denoted by h) of node (x, y), which corresponds to node (x_{ij}, y_{ij}), is given by

$$h = 1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t). \tag{A.9}$$

Then, the unattacked probability (denoted by l) of node (x, y), which corresponds to all recently i-hop attacked nodes, is given by

$$l = \prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)\right). \tag{A.10}$$

Then, the unattacked probability (denoted by s) of node (x, y), which corresponds to all recently attacked nodes, is given by

$$s = \prod_{i=1}^{N}\prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)\right). \tag{A.11}$$

Finally, the probability of node (x, y) being attacked, which corresponds to all recently attacked nodes, is given by

$$C_{(x,y,t)} = 1 - \prod_{i=1}^{N}\prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)\right) \quad (n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots). \tag{A.12}$$

B. APPENDIX B

In intelligent uniform model, the probability of one node being attacked, which is introduced by all new adversaries joined from time $n\tau$, is given by

$$S_{(t)} = 1 - \prod_{i=0}^{m-1}\left(1 - \frac{\lambda\Delta t}{N_g} Q(n\tau + i\Delta t)\right) \quad (n\tau < t < (n+1)\tau,\ n = 0, 1, 2, \ldots). \tag{B.13}$$

Suppose

The number of newly added adversaries follows uniform distribution of time and the time for an adversary to attack a node follows normal distribution which is expressed as Q function. Δt is a very small time period which can be thought of as the smallest time unit in the system; $t = n\tau + m\Delta t$ $(m = 1, 2, 3, \ldots)$; λ is the number of new adversaries that are introduced in a unit time; N_g is the number of current good nodes in the network; $Q(n\tau + (i-1)\Delta t)$ is a normal distribution function; δ is the attack probability in unit time for each node (i.e., a node has δ probability to be chosen as the attacking target in a unit time), which is given by

$$\delta = \frac{\lambda}{N_g}. \tag{B.14}$$

Derivation

In each Δt time period, there are $\lambda\Delta t$ adversaries added to the network. Considering the ith time period which begins from $n\tau + (i-1)\Delta t$ to $n\tau + i\Delta t$, we have what follows.

The probability (denoted by $P_{s,\Delta t}$) of one node being chosen as the attacking target by the new $\lambda\Delta t$ adversaries that are introduced in the ith time period is given by

$$P_{s,\Delta t} = \frac{\lambda\Delta t}{N_g}. \tag{B.15}$$

Then, the probability (denoted by $P_{c,\Delta t}$) of one node being attacked by the new $\lambda\Delta t$ adversaries that are introduced in the ith time period, is given by

$$P_{c,\Delta t} = \frac{\lambda\Delta t}{N_g} Q\big(n\tau + (i-1)\Delta t\big). \tag{B.16}$$

Then, the probability (denoted by $P_{nc,\Delta t}$) of a node that has not been attacked by the new $\lambda\Delta t$ adversaries that are introduced in the ith time period is given by

$$P_{nc,\Delta t} = 1 - \frac{\lambda\Delta t}{N_g} Q\big(n\tau + (i-1)\Delta t\big). \tag{B.17}$$

Thus, the probability (denoted by $P_{nc,\Delta t}$) of a node that has not been attacked by all the new adversaries that are introduced from $n\tau$ to now is given by

$$P_{nc,t} = \prod_{i=1}^{m}\left[1 - \frac{\lambda\Delta t}{N_g} Q\big(n\tau + (i-1)\Delta t\big)\right]. \tag{B.18}$$

Finally, the probability of one node to be attacked, which is introduced by all new adversaries that are introduced from time $n\tau$, is given by

$$S_{(t)} = 1 - \prod_{i=0}^{m-1}\left(1 - \frac{\lambda\Delta t}{N_g} Q(n\tau + i\Delta t)\right). \tag{B.19}$$

REFERENCES

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: scalable coordination in sensor networks,” in *Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’99)*, pp. 263–270, Seattle, Wash, USA, August 1999.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] E. Shi and A. Perrig, “Designing secure sensor networks,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[4] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[5] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[6] C. Jaikaeo, C. Srisathapornphat, and C.-C. Shen, “Diagnosis of sensor networks,” in *Proceedings of the IEEE International Conference on Communications (ICC ’01)*, pp. 1627–1632, Helsinki, Finland, June 2001.

[7] J. Staddon, D. Balfanz, and G. Durfee, “Efficient tracing of failed nodes in sensor networks,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 122–130, Atlanta, Ga, USA, September 2002.

[8] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM ’00)*, pp. 255–265, Boston, Mass, USA, August 2000.

[9] G. Wang, W. Zhang, G. Cao, and T. La Porta, “On supporting distributed collaboration in sensor networks,” in *Proceedings of IEEE Military Communications Conference (MILCOM ’03)*, vol. 2, pp. 752–757, Monterey, Calif, USA, October 2003.

[10] M. Ding, D. Chen, K. Xing, and X. Cheng, “Localized fault-tolerant event boundary detection in sensor networks,” in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’05)*, vol. 2, pp. 902–913, Miami, Fla, USA, March 2005.

[11] B. Krishnamachari and S. Iyengar, “Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks,” *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[12] S. Capkun and J.-P. Hubaux, “Secure positioning of wireless devices with application to sensor networks,” in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’05)*, vol. 3, pp. 1917–1928, Miami, Fla, USA, March 2005.

[13] S. Brands and D. Chaum, “Distance-bounding protocols,” in *Proceedings of Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, pp. 344–359, Lofthus, Norway, May 1993.

[14] L. Lazos and R. Poovendran, “SeRLoc: secure range-independent localization for wireless sensor networks,” in *Proceedings of the ACM Workshop on Wireless Security (WiSe ’04)*, pp. 21–30, Philadelphia, Pa, USA, October 2004.

[15] L. Fang, W. Du, and P. Ning, “A beacon-less location discovery scheme for wireless sensor networks,” in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’05)*, vol. 1, pp. 161–171, Miami, Fla, USA, March 2005.

[16] Crossbow Technology, “MICA2: wireless measurement system,” http://www.xbow.com.

[17] H. Song, L. Xie, S. Zhu, and G. Cao, “Sensor node compromise detection: the location perspective,” in *Proceedings of the International Conference on Wireless Communications and Mobile Computing (IWCMC ’07)*, pp. 242–247, Honolulu, Hawaii, USA, August 2007.

[18] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA ’99)*, pp. 90–100, New Orleans, La, USA, February 1999.

[19] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 197–213, Berkeley, Calif, USA, May 2003.

[20] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS ’03)*, pp. 42–51, Washington, DC, USA, October 2003.

[21] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS ’03)*, pp. 52–61, Washington, DC, USA, October 2003.