20

Distributed Detection of Node Capture Attacks in Wireless Sensor Networks

Jun-Won Ho
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, USA

Abstract

Wireless sensor networks are vulnerable to node capture attacks because sensor nodes are usually deployed in unattended manner. Once attacker captures sensor nodes, he can compromise them and launch various types of attacks with those compromised nodes. Therefore, node capture attacks are hazardous and should be detected as soon as possible to reduce the harm incurred by them. To meet this need, we propose a node capture detection scheme in wireless sensor networks. Our scheme detects the captured sensor nodes by using the sequential analysis. We analytically show that our scheme detects node capture attacks in robust and efficient manner.

1. Introduction

Wireless sensor networks have recently gained much attention in the sense that they can be readily deployed for many different types of missions. In particular, they are useful for the missions that are difficult for humans to carry out. For example, they are suitable for sensing dangerous natural phenomenon such as volcano eruption, biohazard monitoring, and forest fire detection. In addition to these hazardous applications, sensor networks can also be deployed for battle field surveillance, border monitoring, nuclear and chemical attack detection, intrusion detection, flood detection, weather forecasting, traffic surveillance and patient monitoring (Akyildiz et al., 2002).

To carry out a variety of missions, the network operator deploys the base station and a set of small sensor devices in the network field. Specifically, sensor devices form ad-hoc networks, collaborate with each other to sense the phenomenon associated with the assigned missions and then send the sensory data to the base station. The network operator obtains the mission related information by analyzing the data collected at the base station. To help sensor nodes carry out the missions efficiently and effectively, many researchers proposed a variety of the network service and communication protocols (Yick et al., 2008). Specifically, localization, coverage, compression and aggregation protocols have been proposed for the network services. Various network protocols from physical layer to transport layer have been proposed for the communication.

Since sensor networks are often deployed in an unattended manner, most of these protocols are exposed to a variety of attacks such as denial of service attacks, routing disruption and

false data injection attacks, network service disruption attacks (Du & Xiao, 2008; Karlof & Wagner, 2003; Wood & Stankovic, 2002). To defend the wireless sensor networks against these various attacks, many schemes have been developed in the literature. For instance, secure routing schemes have been proposed to mitigate routing disruption attacks (Karlof & Wagner, 2003; Parno et al., 2006). False data injection attacks can be mitigated by using the authentication schemes (Ye et al., 2004; Yu & Li, 2009; Zhu et al., 2004). Secure data aggregation protocols are used to prevent attacker from disrupting aggregation (Chan et al., 2006; Deng et al., 2003; Przydatek et al., 2003; Yang et al., 2006). Many schemes have also been proposed to protect localization and time synchronization protocols from the threat (Capkun & Hubaux, 2006; Ganeriwal et al., 2005; Hu et al., 2008; Li et al., 2005; Liu et al., 2005; Song et al., 2007; KSun et al., 2006).

However, most of them focus on making the protocols be attack-resilient rather than removing the source of attacks. Although attack-resiliency approach mitigates the threats on the network services and communication protocols, this approach requires substantial time and effort to continuously enhance the robustness of the protocols in accordance with the emergence of new types of attacks. Moreover, since it is hard to predict new types of attacks, the protocols will likely have resiliency only after being damaged by new types of attacks. Thus, we need to detect and revoke the sources of attacks as soon as possible to substantially reduce the costs and damages incurred by employing attack-resilience approach. The principle sources of various attacks are compromised sensor nodes in the sense that attacker can compromise sensor nodes by exploiting the unattended nature of wireless sensor networks and thus do any malicious activities with them.

A straightforward strategy for sensor node compromise is to launch *node capture attack* in which adversary physically captures sensor nodes, removes them from the network, compromises and redeploys them in the network. After redeploying compromised nodes, he can mount a variety of attacks with compromised nodes. For example, he can simply monitor a significant fraction of the network traffic that would pass through these compromised nodes. Alternatively, he could jam legitimate signals from benign nodes or inject falsified data to corrupt monitoring operation of the sensors. A more aggressive attacker could undermine common sensor network protocols, including cluster formation, routing, and data aggregation, thereby causing continual disruption to the network operations. Hence, node capture attacks are dangerous and thus should be detected as quickly as possible to minimize the damage incurred by them.

To meet this need, we propose a node capture attack detection scheme in wireless sensor networks. We use the fact that the physically captured nodes are not present in the network during the period from the captured time to redeployed time. Accordingly, captured nodes would not participate in any network operations during that period. By leveraging this intuition, we detect captured nodes by using the Sequential Probability Ratio Test (SPRT) Wald (2004). The main advantage of our scheme is to quickly detect captured nodes with the aid of the SPRT.

The rest of paper is organized as follows. Section 2 describes the network and attacker models. Section 3 describes our node capture attack detection scheme. Section 4 presents the security analysis of our proposed scheme. Section 5 presents the performance analysis of our proposed scheme. Section 6 presents the related work. Finally, Section 7 concludes the paper.

2. Models

In this section, we present the network models and attacker models for our proposed scheme.

2.1 Network Models

We first assume a static sensor network in which the locations of sensor nodes do not change after deployment. We also assume that every sensor node works in promiscuous mode and is able to identify the sources of all messages originating from its neighbors. We believe that this assumption does not incur substantial overhead because each node inspects only the source IDs of the messages from its neighbors rather than the entire contents of the messages.

2.2 Attacker Models

We assume that an attacker can physically capture sensor nodes to compromise them. However, we place limits on the number of sensor nodes that he can physically capture in each target region. This is reasonable from the perspective that an increase in the number of the captured sensor nodes will lead to a rise in the likelihood that attacker is detected by intruder detection mechanisms. Therefore, a rationale attacker will want to physically capture the limited number of sensor nodes in each target region while not being detected by intruder detection mechanisms. Moreover, we assume that it takes a certain amount of time from capturing nodes to redeploying them in the network. This is reasonable in the sense that an attacker needs some time to compromise captured sensor nodes.

3. Node Capture Attack Detection Using the Sequential Probability Ratio Test

In this section, we present the details of node capture detection scheme.

A straightforward approach for node capture detection is to leverage the intuition that a captured node is not present in the network from being captured to being redeployed. Specifically, we first measure the absence time period of a sensor node and then compare it to a pre-defined threshold. If it is more than threshold value, we decide the sensor node as a captured nodes. This simple approach achieves efficient node capture detection capability as long as a threshold value is properly configured. However, it is not easy to configure a proper a threshold value to detect captured nodes. If we set threshold to a high value, it is likely that captured nodes bypass the detection. On the contrary, if we set threshold to a low value, it is likely that benign nodes can be detected as captured nodes. To minimize these false positives and negatives, we need to set up threshold in such a way that it is dynamically changed in accordance with the measured absence time duration for a node. To meet this need, we use the Sequential Probability Ratio Test (SPRT) (Wald, 2004), which is a statistical decision process and is regarded as a dynamic threshold scheme (Jung et al., 2004). We can take advantage of using the SPRT from the perspective that the SPRT reaches a decision with few pieces of samples while achieving low false positive and false negative rates (Wald, 2004). Specifically, we apply the SPRT to node capture detection problem as follows. For each time slot, every sensor node measures the number of messages sent by its neighbors. Each time the number of messages sent by a neighbor is above (resp. equal to) zero, it will expedite the test process to accept the null (resp. alternate) hypothesis that the neighbor is present (resp. absent) in the network. Once a node accepts alternate hypothesis, it decides that the neighbor has been captured and disconnects the communication with the neighbor.

After deployment, every sensor node u discovers its neighboring nodes. The entire time domain of node u is divided into a series of time slots. For each neighbor node v, node u measures the number of messages sent by v every time slot. We denote the number of messages whose originator is v during the ith time slot by N_i. Let V_i be denote a Bernoulli random

variable that is defined as:

$$V_i = \begin{cases} 1 & \text{if } N_i = 0 \\ 0 & \text{if } N_i > 0 \end{cases} \quad (1)$$

where $i \geq 1$. The success probability δ of Bernoulli distribution is defined as

$$Pr(V_i = 1) = 1 - Pr(V_i = 0) = \delta. \quad (2)$$

If δ is smaller than or equal to a preset threshold δ', it is likely that node v is present in the network and is accordingly not captured by attacker. On the contrary, if $\delta > \delta'$, it is likely that node v is absent in the network and is accordingly captured by attacker. The problem of deciding whether v is captured or not can be formulated as a hypothesis testing problem with null and alternate hypotheses of $\delta \leq \delta'$ and $\delta > \delta'$, respectively. In this problem, we need to devise an appropriate sampling strategy in order to prevent hypothesis testing from leading to a wrong decision. In particular, we should specify the maximum possibilities of wrong decisions that we want to tolerate for a good sampling strategy. To do this, we reformulate the above hypothesis testing problem as one with null and alternate hypotheses of $\delta \leq \delta_0$ and $\delta \geq \delta_1$, respectively, such that $\delta_0 < \delta_1$. In this reformulated problem, the acceptance of the alternate hypothesis is regarded as a false positive error when $\delta \leq \delta_0$, and the acceptance of the null hypothesis is regarded as false negative error when $\delta \geq \delta_1$. To prevent the decision process from making these two types of errors, we define a user-configured false positive α' and false negative β' in such a way that the false positive and negative should not exceed α' and β', respectively.

Now we present how node u performs the SPRT to make a decision of v with the n observed samples, where N_i is treated as a sample. Let us define H_0 as the null hypothesis that v is present in the network and is not captured by attacker, H_1 as the alternate hypothesis that v is not present in the network and is captured by attacker. We then define L_n as the log-probability ratio on n samples, given as:

$$L_n = \ln \frac{\Pr(V_1, \ldots, V_n | H_1)}{\Pr(V_1, \ldots, V_n | H_0)}$$

Assume that V_i is independent and identically distributed. Then L_n can be rewritten as:

$$L_n = \ln \frac{\prod_{i=1}^{n} \Pr(V_i | H_1)}{\prod_{i=1}^{n} \Pr(V_i | H_0)} = \sum_{i=1}^{n} \ln \frac{\Pr(V_i | H_1)}{\Pr(V_i | H_0)} \quad (3)$$

Let y_n denote the number of times that $V_i = 1$ in the n samples. Then we have $L_n = y_n \ln \frac{\delta_1}{\delta_0} + (n - y_n) \ln \frac{1-\delta_1}{1-\delta_0}$ where $\delta_0 = \Pr(V_i = 1 | H_0)$, $\delta_1 = \Pr(V_i = 1 | H_1)$. The rationale behind the configuration of δ_0 and δ_1 is as follows. δ_0 should be configured in accordance with the likelihood of the occurrence that a benign node is determined to be absent in the network during a time slot. δ_1 should be configured to consider the likelihood of the occurrence that a captured node is determined to be absent in the network during a time slot. On the basis of the log-probability ratio L_n, the SPRT for H_0 against H_1 is given as follows:

- $L_n \leq \ln \frac{\beta'}{1-\alpha'}$: accept H_0 and terminate the test.
- $L_n \geq \ln \frac{1-\beta'}{\alpha'}$: accept H_1 and terminate the test.
- $\ln \frac{\beta'}{1-\alpha'} < L_n < \ln \frac{1-\beta'}{\alpha'}$: continue the test process with another observation.

This SPRT can be written as:

- $y_n \leq s_0(n)$: accept H_0 and terminate the test.
- $y_n \geq s_1(n)$: accept H_1 and terminate the test
- $s_0(n) < y_n < s_1(n)$: continue the test process with another observation.

Where

$$s_0(n) = \frac{\ln \frac{\beta'}{1-\alpha'} + n \ln \frac{1-\delta_0}{1-\delta_1}}{\ln \frac{\delta_1}{\delta_0} - \ln \frac{1-\delta_1}{1-\delta_0}}, \quad s_1(n) = \frac{\ln \frac{1-\beta'}{\alpha'} + n \ln \frac{1-\delta_0}{1-\delta_1}}{\ln \frac{\delta_1}{\delta_0} - \ln \frac{1-\delta_1}{1-\delta_0}}$$

,α' and β' are the user-configured false positive and false negative rates, respectively.
If the SPRT terminates in acceptance of H_0, node u restarts the SPRT with newly received messages from v. However, if the SPRT accepts H_1, u terminates the SPRT on v, decides v as a captured node, and disconnects the communication with v.
The pseudocode for the SPRT is presented as Algorithm 1.

Algorithm 1 SPRT for replica detection

```
INITIALIZATION: t = 1, y = 0
INPUT: N_t
OUTPUT: accept the hypothesis H_0 or H_1
compute s_0(t) and s_1(t)
if N_t == 0 then
   y = y + 1
end if
if y >= s_1(t) then
   accept the alternate hypothesis H_1 and terminate the test
end if
if y <= s_0(t) then
   accept the null hypothesis H_0 and initialize t to 1 and y to 0
   return;
end if
t = t + 1
```

4. Security Analysis

In this section, we first present the detection capability of our scheme and then discuss about the limitations of node capture attacks under the presence of our scheme and countermeasures against some possible attack strategies against our scheme.
In the SPRT, the following types of errors are defined.

- α : error probability that the SPRT leads to accepting H_1 when H_0 is true.
- β : error probability that the SPRT leads to accepting H_0 when H_1 is true.

Since H_0 is the hypothesis that a node u has not been captured, α and β are the false positive and false negative probabilities of the SPRT, respectively. According to Wald's theory (Wald, 2004), the upper bounds of α and β are:

$$\alpha \leq \frac{\alpha'}{1-\beta'}, \qquad \beta \leq \frac{\beta'}{1-\alpha'} \tag{4}$$

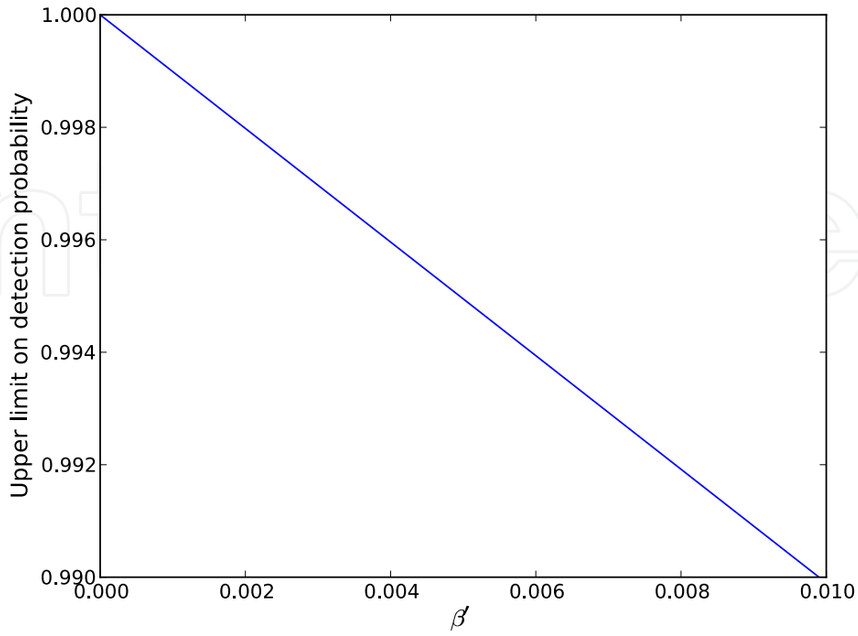


Fig. 1. Upper limit on detection probability vs. β' when $\alpha' = 0.01$.

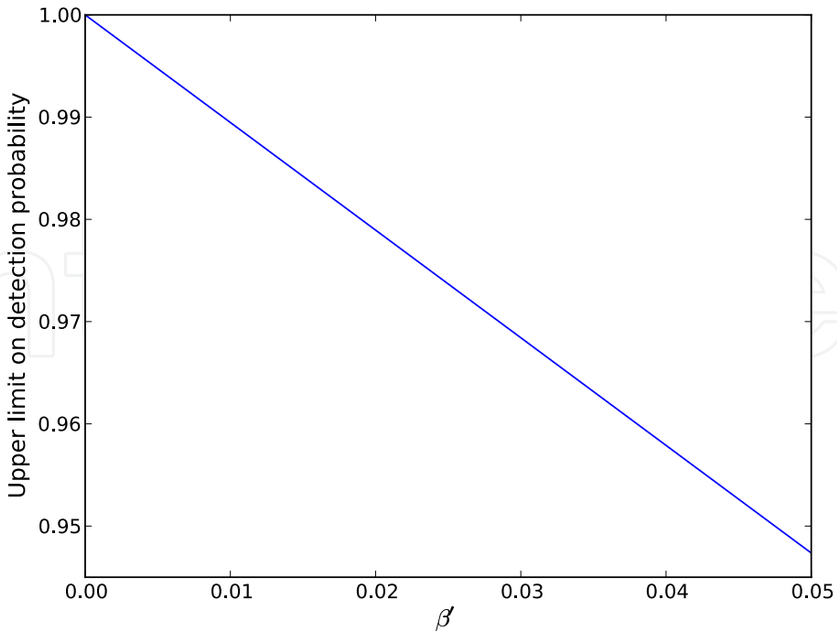


Fig. 2. Upper limit on detection probability vs. β' when $\alpha' = 0.05$.

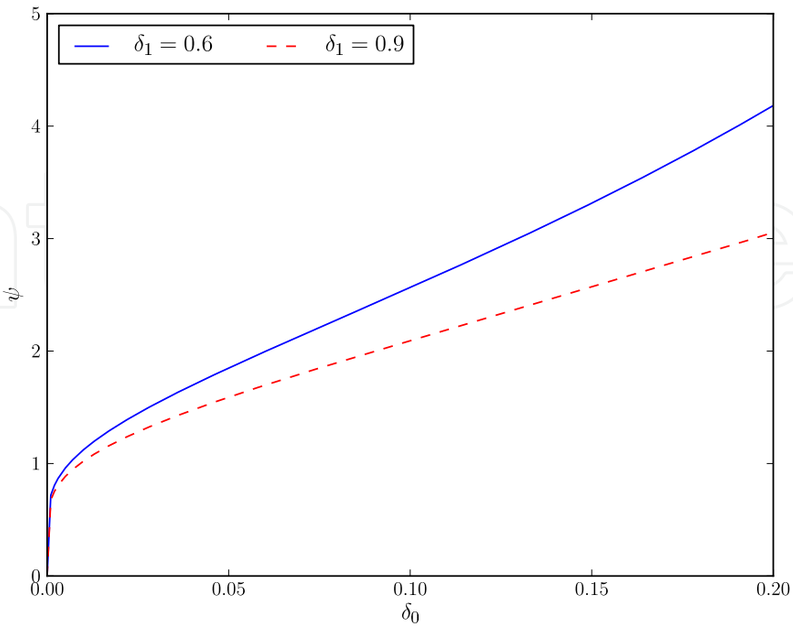


Fig. 3. ψ vs. δ_0 when $\alpha' = \beta' = 0.01$.

Furthermore, Wald proved that the sum of the false positive and negative probabilities of the SPRT are limited by the sum of user-configured false positive and negative probabilities. Namely, the following inequality holds:

$$\alpha + \beta \leq \alpha' + \beta' \tag{5}$$

Since β is the false negative probability, $(1 - \beta)$ is the node capture detection probability. Accordingly, the lower bound on the node catpure detection probability will be:

$$(1 - \beta) \geq \frac{1 - \alpha' - \beta'}{1 - \alpha'} \tag{6}$$

From Equations 4 and 6, we can see that low user-configured false positive and negative probabilities will lead to a low false negative probability for the sequential test process. Hence, it will result in high detection rates.

As shown in Figures 1 and 2, we study how α' and β' affect the upper limit of node capture detection probability $(1 - \beta)$. Specifically, the upper limit decreases as the rise in β' when the user configures α' to 0.01 and 0.05. However, we see that the upper limit is bounded from below 0.99 (resp., 0.945) when $\alpha' = 0.01$ (resp., 0.05) as long as β' is configured to at most 0.01 (resp., 0.05). Hence, the node capture detection capability is guaranteed with at least probability of 0.945 when both α' and β' are set to at most 0.05.

Now we derive the limitation of the time period from when a node is captured and removed in location L to when it is redeployed in the same location L. Suppose that the entire n time slots are taken from the removal to redeployment of captured node. Since the captured node

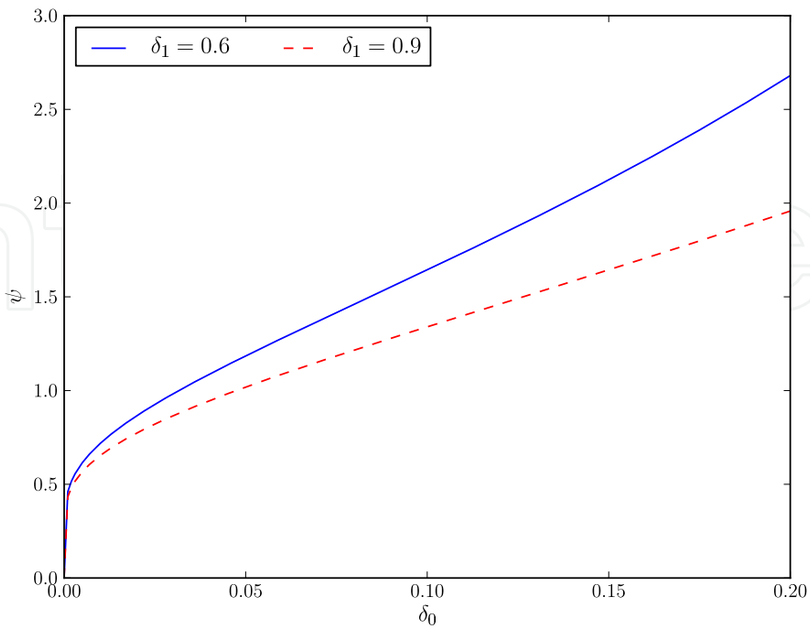


Fig. 4. ψ vs. δ_0 when $\alpha' = \beta' = 0.05$.

will not be present in the network for n time slots and a time slot corresponds to a sample in the SPRT, $y_n = n$ holds. Accordingly, $y_n = n < s_1(n)$ should hold for captured node to avoid being detected. In other words, the following Inequality should hold to bypass the detection:

$$n < \psi = \frac{\ln \frac{1-\beta'}{\alpha'}}{\ln \frac{\delta_1}{\delta_0}} \tag{7}$$

As shown in Figures 3 and 4, we study how the values of δ_0 and δ_1 affect ψ when $\alpha' = 0.01, \beta' = 0.01$ and $\alpha' = 0.05, \beta' = 0.05$. Specifically, ψ increases as δ_0 rises when δ_1 is configured to 0.6 and 0.9, but it decreases as δ_1 rises when δ_0 is fixed. We see from this that small and large values of δ_0 and δ_1 lead to the small value of ψ. We also observe that n is less than 5 and 3 in the case of $\alpha' = \beta' = 0.01$ and $\alpha' = \beta' = 0.05$, respectively. This means that attacker should finish compromising and redeploying the captured node within at most five time slots in order to prevent them from being detected. Hence, our scheme will substantially limit the time duration for captured node not to be detected.

However, if a captured node is not redeployed in its initial location L but in different location L', even though it cannot be accepted as legitimate neighbors by the nodes around L, it can still be accepted as legitimate neighbors by the nodes around L' and thus have an impact on these nodes. To defend the network against this attack, we propose a countermeasure based on the group deployment strategy. This involves three important assumptions.

First, we assume that sensor nodes are deployed in group-by-group. More specifically, sensor nodes are grouped together by the network operator and programmed with the corresponding group information before deployment, with each group of nodes being deployed towards the same location, called the *group deployment point*. After deployment, the group members exhibit similar geographic relations. We argue that this is reasonable for sensor network in

which nodes are spread over a field, such as being dropped from an airplane or spread out by hand. A simple way to do this would be to keep the groups of nodes in bags marked with the group IDs and use a marked map with the group IDs on it. All that is needed is a map of the territory and a way to pre-determine the deployment points, such as assigning a point on a grid to each group. This argument is further supported by the fact that the group deployment strategy has been used for various applications in sensor networks such as key distribution (Du et al., 2004), detection of anomalies in localization (Du et al., 2005), and public key authentication (Du et al., 2005).

The deployment follows a particular probability density function (pdf), say f, which describes the likelihood of a node being a certain distance from its group deployment point. For simplicity, we use a two-dimensional Gaussian distribution to model f, as in (Du et al., 2005). Let (x_g, y_g) be the group deployment point for a group g. A sensor node in group g is placed in a location (x, y) in accordance with the following model:

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_g)^2+(y-y_g)^2}{2\sigma^2}} \tag{8}$$

where (x, y) is group deployment point and σ is the standard deviation of the two-dimensional Gaussian distribution. According to Equation 8, 68% and 99% of nodes in a group are placed within a circle whose center is the group deployment point and radius is σ and 3σ, respectively.

Second, we assume that it takes some time for an attacker to capture and compromise a sensor node. This need not be a long time, but we assume that there is a minimum amount of time that it takes to compromise a node once it has been deployed. [1] Third, we assume that the clocks of all nodes are loosely synchronized with a maximum error of ϵ. This can be achieved by the use of secure time synchronization protocols as proposed in (Ganeriwal et al., 2005; Hu et al., 2008; Song et al., 2007; KSun et al., 2006).

Under these assumptions, the main idea of the proposed countermeasure is to pre-announce the deployment time of each group, and have nodes treat as captured and redeployed any node that initiates communications after a long time of its expected deployment. More specifically, when a group G_u of nodes are deployed, they will be pre-loaded with a time stamp T_u that is digitally signed by a trusted server. This time stamp indicates that the sensor nodes in G_u should finish neighbor discovery before time T_u. If they try to setup neighbor connections with other nodes after time T_u, they are considered to be captured and redeployed nodes. The time stamp T_u should be a function of the deployment time T, the time T_r needed for capturing, compromising, and redeploying a node, and the maximum time synchronization error ϵ. Specifically, the network operator should set $T + T_d + \epsilon < T_u < T + T_d + T_r - \epsilon$, where T_d is the neighbor discovery time, such that no nodes should have clocks too fast to accept the new node, but no new node could be compromised and accepted in time. This means that $\epsilon < 0.5T_c$ determines the maximum amount of allowable error.

5. Performance Analysis

This section describes how many observations are required on average for each node to decide whether its neighboring node has been captured or not.

Let n denote the number of samples to terminate the SPRT. Since n is changed with the types of samples, it is treated as a random variable with an expected value $E[n]$. According to (Wald,

[1] According to (Hartung et al., 2005), it took approximately one minute to compromise a node.

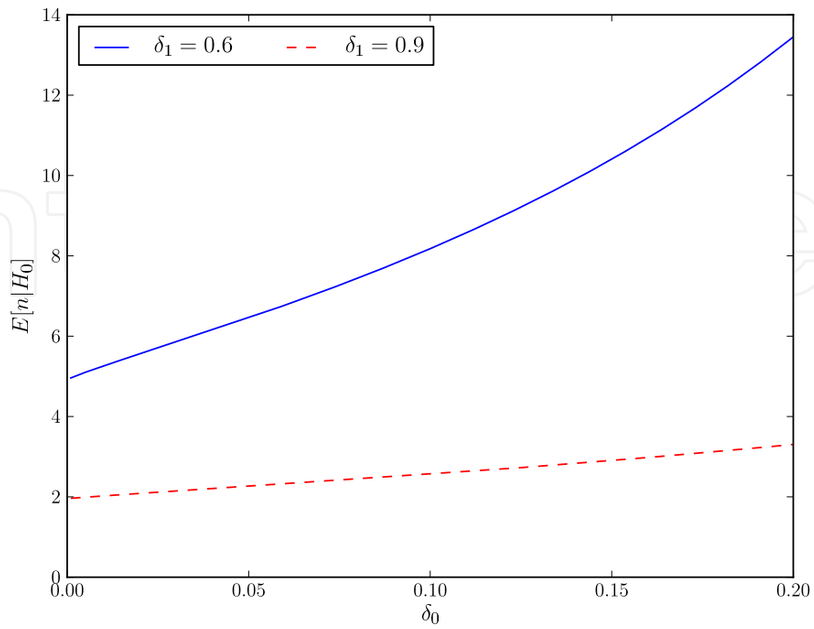


Fig. 5. $E[n|H_0]$ vs. δ_0 when $\alpha' = \beta' = 0.01$.

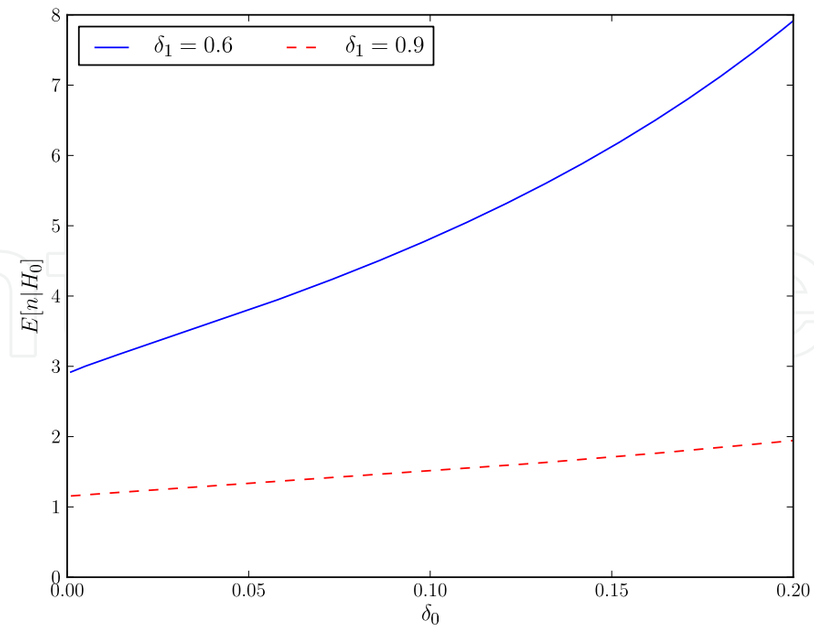


Fig. 6. $E[n|H_0]$ vs. δ_0 when $\alpha' = \beta' = 0.05$.

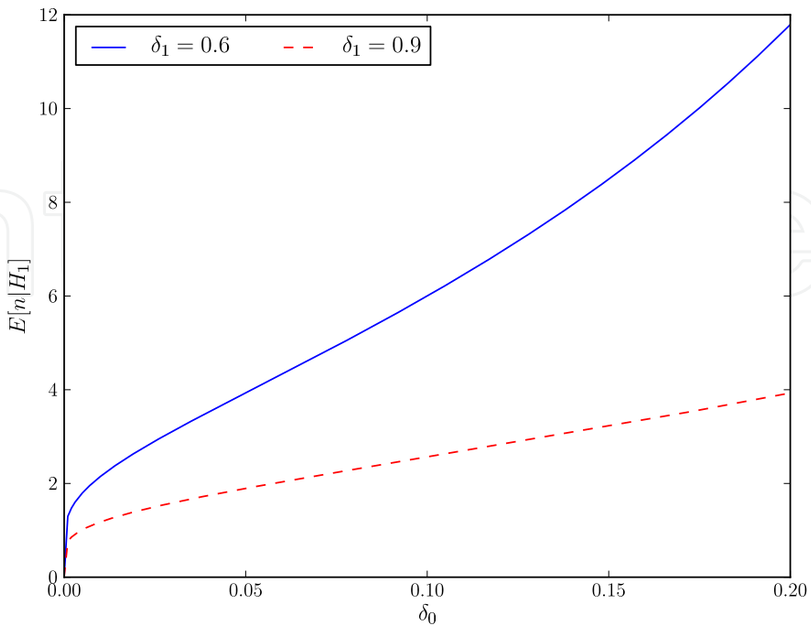


Fig. 7. $E[n|H_1]$ vs. δ_0 when $\alpha' = \beta' = 0.01$.

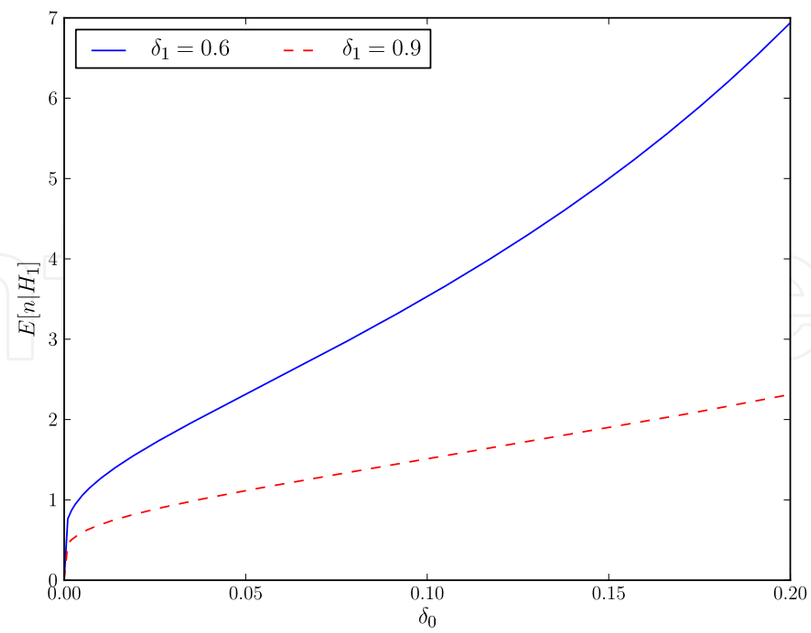


Fig. 8. $E[n|H_1]$ vs. δ_0 when $\alpha' = \beta' = 0.05$.

2004), $E[n]$ is given by:

$$E[n] = \frac{E[L_n]}{E\left[\ln \frac{\Pr(V_i|H_1)}{\Pr(V_i|H_0)}\right]} \tag{9}$$

From Equation 9, we compute the expected values of n conditioned on hypotheses H_0 and H_1 as follows:

$$\begin{aligned} E[n|H_0] &= \frac{(1-\alpha')\ln\frac{\beta'}{1-\alpha'} + \alpha'\ln\frac{1-\beta'}{\alpha'}}{\delta_0 \ln\frac{\delta_1}{\delta_0} + (1-\delta_0)\ln\frac{1-\delta_1}{1-\delta_0}} \\ E[n|H_1] &= \frac{\beta'\ln\frac{\beta'}{1-\alpha'} + (1-\beta')\ln\frac{1-\beta'}{\alpha'}}{\delta_1 \ln\frac{\delta_1}{\delta_0} + (1-\delta_1)\ln\frac{1-\delta_1}{1-\delta_0}} \end{aligned} \tag{10}$$

As shown in Figures 5, 6, 7, and 8, we study how the values of δ_0 and δ_1 affect $E[n|H_0]$ and $E[n|H_1]$ when $\alpha' = \beta' = 0.01$ and $\alpha' = \beta' = 0.05$. Specifically, $E[n|H_1]$ increases as the rise of δ_0 for a given value of δ_1. This means that captured nodes are detected with a small number of samples when δ_0 is small. For a given value of δ_0, $E[n|H_1]$ decreases as the increase of δ_1. This means that large values of δ_1 reduce the number of samples required for node capture detection. Similarly, the small value of δ_0 and the large value of δ_1 contribute to decrease of $E[n|H_0]$, leading to the small number of samples required for deciding that benign node is not captured.

6. Related Work

In this section, we describe a number of research works that are related to node capture detection in wireless sensor networks.

In (Tague & Poovendran, 2008), node capture attacks are modeled in wireless sensor networks. However, this work did not propose detection schemes against node capture attacks. In (Conti et al., 2008), node capture attack detection scheme was proposed in mobile sensor networks. They leverage the intuition that a mobile node is regarded as being captured if it is not contacted by other mobile nodes during a certain period of time. However, this scheme will not work in static sensor networks where sensor nodes do not move after deployment.

Software-attestation based schemes have been proposed to detect the subverted software modules of sensor nodes (Park & Shin, 2005; Seshadri et al., 2004; Shaneck et al., 2005; Yang et al., 2007). Specifically, the base station checks whether the flash image codes have been maliciously altered by performing attestation randomly chosen portions of image codes or the entire codes in (Park & Shin, 2005; Seshadri et al., 2004; Shaneck et al., 2005). In (Yang et al., 2007), a sensor node's image codes are attested by its neighbors. However, all these schemes require each sensor to be periodically attested and thus incur a large overhead in terms of communication and computation.

Reputation-based trust management schemes have been proposed to manage individual node's trust in accordance with its actions (Ganeriwal & Srivastava, 2004; Li at al., 2007; YSun et al., 2006). Specifically, a reputation-based trust management scheme was proposed in (Ganeriwal & Srivastava, 2004). The main idea of the scheme is to use a Bayesian formulation in order to compute an individual node's trust. In (YSun et al., 2006) information theoretic frameworks for trust evaluation were proposed. Specifically, entropy-based and probability-based schemes have been proposed to compute an individual node's trust. In (Li at al., 2007), node mobility is leveraged to reduce an uncertainty in trust computation and speed up the trust convergence. However, these trust management schemes do not revoke compromised

nodes and thus compromised nodes can keep performing malicious activities in the network. ID traceback schemes have been proposed to locate the malicious source of false data (Ye et al., 2007; Zhang et al., 2006). However, they only trace a source of the data sent to the base station and thus they do not locate the malicious sources that send false data or control messages to other benign nodes in the network.

After physically capturing and compromising a few sensor nodes, attacker can generate many replica nodes with the same ID and secret keying materials as the compromised nodes, and mount a variety of attacks with replica nodes. Randomized and line-selected multicast schemes were proposed to detect replicas in wireless sensor networks (Parno et al., 2005). In the randomized multicast scheme, every node is required to multicast a signed location claim to randomly chosen witness nodes. A witness node that receives two conflicting location claims for a node concludes that the node has been replicated and initiates a process to revoke the node. The line-selected multicast scheme reduces the communication overhead of the randomized multicast scheme by having every claim-relaying node participate in the replica detection and revocation process.

A Randomized, Efficient, and Distributed (RED) protocol was proposed to enhance the line-selected multicast scheme of (Parno et al., 2005) in terms of replica detection probability, storage and computation overheads (Conti et al., 2007). However, RED still has the same communication overhead as the line-selected multicast scheme of (Parno et al., 2005). More significantly, their protocol requires repeated location claims over time, meaning that the cost of the scheme needs to be multiplied by the number of runs during the total deployment time. Localized multicast schemes based on the grid cell topology detect replicas by letting location claim be multicasted to a single cell or multiple cells (Zhu et al., 2007). The main strength of (Zhu et al., 2007) is that it achieves higher detection rates than the best scheme of (Parno et al., 2005). However, (Zhu et al., 2007) has similar communication overheads as (Parno et al., 2005).

A clone detection scheme was proposed in sensor networks (Choi et al., 2007). In this scheme, the network is considered to be a set of non-overlapping subregions. An exclusive subset is formed in each subregion. If the intersection of subsets is not empty, it implies that replicas are included in those subsets. Fingerprint-based replica node detection scheme was proposed in sensor networks (Xing et al., 2008). In this scheme, nodes report fingerprints, which identify a set of their neighbors, to the base station. The base station performs replica detection by using the property that fingerprints of replicas conflict each other.

7. Conclusion

In this paper, we proposed a node capture attack detection scheme using the Sequential Probability Ratio Test (SPRT). We showed the limitations of the benefits that attacker can take from launching node capture attacks when our scheme is employed. We also analytically showed that our scheme detects node capture attacks with a few number of samples while sustaining the false positive and false negative rates below 1%.

8. References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks : a survey. *Computer Networks* 38(4):393–422, March 2002.

Boneh, D. & Franklin, M.K. (2001). Identity-based encryption from the weil pairing. In *CRYPTO*, pages:213-229, August 2001.

Capkun, S. & Hubaux, J.P. (2006). Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):221–232, February 2006.

Chan, H., Perrig, A., & Song, D. (2003). Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pages:197-213 , May 2003.

Chan, H., Perrig, A., & Song, D. (2006). Secure hierarchical in-network aggregation in sensor networks . In *ACM CCS*, pages:278-287, October 2006.

Cocks, C. (2001). An identity based encryption scheme based on quadratic residues. In *IMA International Conference on Cryptography and Coding*, pages:360-363, December 2001.

Choi, H., Zhu, S., & La Porta, T.F. (2007). {SET}: detecting node clones in sensor networks. In *IEEE/CreateNet SecureComm*, pages:341-350, September 2007.

Conti, M., Pietro, R.D., Mancini, L.V., & Mei, A. (2007). A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks. In *ACM Mobihoc*, pages:80-89, September 2007.

Conti, M., Pietro, R., Mancini, L., & Mei, A. (2008). Emergent Properties: Detection of the Node-capture Attack in Mobile Wireless Sensor Networks. In *ACM WiSec*, April 2008.

Delgosha, F. & Fekri, F. (2006). Threshold key-establishment in distributed sensor networks using a multivariate scheme. In *IEEE INFOCOM*, pages:1-12, April 2006.

Deng, J., Han, R., & Mishra, S. (2003). Security support for in-network processing in wireless sensor networks. In *ACM SASN*, pages:83-93, October 2003.

Du, W., Deng, J., Han, Y. S., & Varshney, P. (2003). A pairwise key pre-distribution scheme for wireless sensor networks. In *ACM CCS*, pages 42–51, October 2003.

Du, W., Deng, J., Han, Y. S., Chen, S., & Varshney, P. (2004). A key management scheme for wireless sensor networks using deployment knowledge. In *IEEE INFOCOM*, pages:586-597, March 2004.

Du, W., Fang, L., & Ning, P. (2005). {LAD}: localization anomaly detection for wireless sensor networks. In *IEEE IPDPS*, pages:874-886, April 2005.

Du, W., Wang, R., & Ning, P. (2005). An efficient scheme for authenticating public keys in sensor networks. In *ACM MobiHoc*, pages:58-67, May 2005.

Du, X. & Xiao, Y. (2008). Chapter 17: A survey on sensor network security *Springer Wireless Sensor Networks and Applications*, 2008

Eschenauer, L. & Gligor, V. (2002). A key-management scheme for distributed sensor networks. In *ACM CCS*, pages:41-47, November 2002.

Ganeriwal, S.& Srivastava, M. (2004). Reputation-based framework for high integrity sensor networks. In *ACM SASN*, pages:66-77, October 2004.

Ganeriwal, S., Čapkun, S., Han, C.C., & Srivastava, M.B. (2005). Secure time synchronization service for sensor networks. In *ACM WiSe*, pages:97-106, September 2005.

Gupta, V., Millard, M., Fung, S., Zhu, Y., Gura, N., and Eberle, S., & Chang, H. (2005). Sizzle: a standards-based end-to-end security architecture for the embedded internet. In *IEEE PerCom*, pages:247-256, March 2005.

Hartung, C., Balasalle, J., & Han, R. (2005). Node compromise in sensor networks: the need for secure systems. In *Technical Report CU-CS-990-05, Department of Computer Science, University of Colorado at Boulder*, January 2005.

Hu, L. & Evans, D. (2003). Using directional antennas to prevent wormhole attacks. In *Proceedings of the 11th Network and Distributed System Security Symposium*, pages 131–141, February 2003.

Hu, Y.C., Perrig, A., & Johnson, D.B. (2003). Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of INFOCOM 2003*, April 2003.

Hu, X., Park, T., & Shin, K. G. (2008). Attack-tolerant time-synchronization in wireless sensor networks. In *IEEE INFOCOM*, pages:41-45, April 2008.

Jung, J., Paxon, V., Berger, A.W. & Balakrishnan, H. (2004). Fast port scan detection using sequential hypothesis testing. In *IEEE Symposium on Security and Privacy*, pages:211-225, May 2004.

Karlof, C. & Wagner, D. (2003). Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks Journal*, 1(2-3):293-315, September 2003.

Li, Z., Trappe, W., Zhang, Y., & Nath, B. (2005). Robust statistical methods for securing wireless localization in sensor networks. In *IEEE IPSN*, pages:91-98, April 2005.

Li, F., & Wu., J. (2007). Mobility reduces uncertainty in {MANET}. In *IEEE INFOCOM*, pages:1946-1954, May 2007.

Liu, A. & Ning, P. (2008). TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks. In *IEEE IPSN*, pages:245-256, April 2008.

Liu, D. & Ning, P. (2003). Establishing pariwise keys in distributed sensor networks. In *ACM CCS*, pages:52-61, October 2003.

Liu, D., Ning, P., & Du, W. (2005). Attack-resistant location estimation in sensor networks. In *IEEE IPSN*, pages:99-106, April 2005.

Malan, D., Welsh, M., & Smith, M. (2004). A public-key infrastructure for key distribution in tinyOS based on elliptic curve cryptography. In *IEEE SECON*, pages:71-80, October 2004.

Park, T. & Shin, K. G. (2005). Soft tamper-proofing via program integrity verification in wireless sensor networks. In IEEE Trans. Mob. Comput., 4(3):297-309, 2005.

Parno, B., Perrig, A., and Gligor, V.D. (2005). Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, pages:49-63, May 2005.

Parno, B., Luk, M., Gaustad, E., and Perrig, A. (2006). Secure sensor network routing: a cleanslate approach. In *ACM CoNEXT*, December 2006.

Przydatek, B., Song, D., & Perrig, A. (2003). {SIA}: secure information aggregation in sensor networks. In *ACM SenSys*, pages:69-102, November 2003.

Seshadri, A., Perrig, A., van Doorn, L., & Khosla, P. (2004). {SWATT}: softWare-based attestation for embedded devices. In *IEEE Symposium on Security and Privacy*, pages:272-282, May 2004.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages:47-53, August 1984.

Shaneck, M., Mahadevan, K., Kher, V., & Kim, Y. (2005). Remote software-based attestation for wireless sensors. In *ESAS*, July 2005.

Song, H., Zhu, S., & Cao, G. (2007). Attack-resilient time synchronization for wireless sensor networks. *Ad Hoc Networks*, 5(1):112–125, January 2007.

Sun, K., Ning, P., Wang, C., Liu, A., & Zhou, Y. (2006). TinySeRSync: secure and resilient time synchronization in wireless sensor networks. In *ACM CCS*, pages:264-277, 2006.

Sun, Y., Han, Z., Yu, W., & Liu, K. (2006). A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks. In *IEEE INFOCOM*, pages:1-13, April 2006.

Tague, P. & Poovendran, R. (2008). Modeling node capture attacks in wireless sensor networks. In *Allerton Conference on Communication, Control, and Computing* , September 2008.

Wald, A. (2004). Sequential analysis. *Dover Publications*, 2004.

Wang, H., Sheng, B., Tan, C.C., & Li, Q. (2008). Comparing symmetric-key and public-key based security schemes in sensor networks: a case study of user access control. In *IEEE ICDCS*, pages:11-18, 2008.

Wood, A. D. & Stankovic, J. A. (2002). Denial of service in sensor networks. *IEEE Computer* 35(10):54–62, 2002

Xing, K., Liu, F., Cheng, X., & Du, H.C. (2008). Real-time detection of clone attacks in wireless sensor networks. In *IEEE ICDCS*, pages:3-10, June 2008.

Yang, Y., Wang, X., Zhu, S., & Cao, G. (2006). {SDAP}: a secure hop-by-hop data aggregation protocol for sensor networks. In *ACM MOBIHOC*, 2006.

Yang, Y., Wang, X., Zhu, S., & Cao, G. (2007). Distributed software-based attestation for node compromise detection in sensor networks. In *IEEE SRDS*, pages:219-230, October 2007.

Ye, F., Luo, H., Lu, S., & Zhang, L. (2004). Statistical en-route filtering of injected false data in sensor networks. In *IEEE INFOCOM*, 2004.

Ye, F., Yang, H., & Liu, Z. (2007). Catching moles in sensor networks. In *IEEE ICDCS*, June 2007.

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, August 2008.

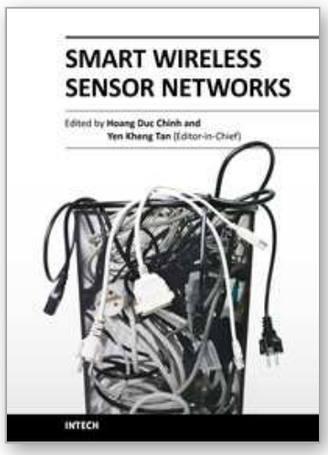
Yu, L. & Li, J. (2009). Grouping-based resilient statistical en-route filtering for sensor networks. To appear in *IEEE INFOCOM*, April 2009.

Zhang, Y., Yang, J., Jin, L., & Li, W. (2006). Locating compromised sensor nodes through incremental hashing authentication. In *DCOSS*, June 2006.

Zhang, W., Tran, M., Zhu, S., & Cao, G. (2007). A random perturbation-based scheme for pairwise key establishment in sensor networks. In *ACM Mobihoc*, pages:90-99, September 2007.

Zhu, S., Setia, S., Jajodia, S., & Ning, P. (2004). An interleaved by hop-by-hop authentication scheme for filtering injected false data in sensor networks. In *IEEE Symposium on Security and Privacy*, pages:259-271, May 2004.

Zhu, B., Addada, V.G.K., Setia, S., Jajodia, S., & Roy, S. (2007). Efficient distributed detection of node replication attacks in sensor networks. In *ACSAC*, pages:257-267, December 2007.



Smart Wireless Sensor Networks

Edited by Yen Kheng Tan

ISBN 978-953-307-261-6

Hard cover, 418 pages

Publisher InTech

Published online 14, December, 2010

Published in print edition December, 2010

The recent development of communication and sensor technology results in the growth of a new attractive and challenging area â€“ wireless sensor networks (WSNs). A wireless sensor network which consists of a large number of sensor nodes is deployed in environmental fields to serve various applications. Facilitated with the ability of wireless communication and intelligent computation, these nodes become smart sensors which do not only perceive ambient physical parameters but also be able to process information, cooperate with each other and self-organize into the network. These new features assist the sensor nodes as well as the network to operate more efficiently in terms of both data acquisition and energy consumption. Special purposes of the applications require design and operation of WSNs different from conventional networks such as the internet. The network design must take into account of the objectives of specific applications. The nature of deployed environment must be considered. The limited of sensor nodesâ€™ resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design. A smart wireless sensor network must be able to deal with these constraints as well as to guarantee the connectivity, coverage, reliability and security of networkâ€™s operation for a maximized lifetime. This book discusses various aspects of designing such smart wireless sensor networks. Main topics includes: design methodologies, network protocols and algorithms, quality of service management, coverage optimization, time synchronization and security techniques for sensor networks.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821

IntechOpen