Adaptive Network Defense Management for Countering Smart Attack and Selective Capture in Wireless Sensor Networks

Hamid Al-Hamadi and Ing-Ray Chen, *Member, IEEE*

***Abstract*—We propose and analyze adaptive network defense management for countering smart attack and selective capture which aim to cripple the basic data delivery functionality of a base station based wireless sensor network. With selective capture, the adversaries strategically capture sensors and turn them into inside attackers. With smart attack, an inside attacker is capable of performing random, opportunistic and insidious attacks to evade detection and maximize their chance of success. We develop a model-based analysis methodology with simulation validation to identify the best defense protocol settings under which the sensor network lifetime is maximized against selective capture and smart attack.**

Index Terms— **Wireless sensor networks, selective capture, smart attack, multipath routing, intrusion tolerance, intrusion detection, MTTF.**

1 Introduction

In a base station based wireless sensor network (WSN), data packets have to be forwarded to the base station (BS) via multi-hop routing using sensor nodes (SNs) as relays. SNs close to the BS (called critical SNs) are desirable targets for capture attack since compromised SNs close to the BS can best intercept data packets forwarded to the BS to disrupt the basic data delivery functionality. In the literature, various schemes have been designed for preserving critical SNs from energy exhaustion so as to prolong the system lifetime; however, how to counter selective capture, i.e., critical SNs are targets of selective capture attacks, is an open issue [1]. Once a node is captured and turned into a malicious node, it becomes an inside attacker. Modeling smart attacker behaviors and studying their effects on security is little explored in the literature and is another open issue [2].

In this paper, we propose and analyze adaptive network defense management for countering smart attack and selective capture which aim to cripple the basic data delivery functionality of a wireless sensor network. With selective capture, the adversaries strategically capture sensors and turn them into inside attackers. With smart attack, an inside attacker is capable of performing random, opportunistic and insidious attacks to evade detection and maximize their chance of success.

From the data flow perspective, WSNs can be classified as source-driven or query-based. In a source-driven WSN, SNs sense the environment at a fixed rate and periodically transmit sensing data to the BS. In a query-based WSN, a query is issued by the BS proactively or reactively, and SNs in the feature areas collect data and forward data to the BS in response to the query. This paper focuses on query-based WSNs. There is a wide range of query-based WSN applications to which the proposed adaptive network defense management for countering smart attack and selective capture can be applied, including:

- Oil and gas [43], [44]: A query-based WSN with SNs monitoring noise, vibration, humidity, electrical characteristics, temperature, radiation, toxic gases, etc. and reporting sensed data to the BS upon inquiry.
- Nuclear power plants [45]: A query-based WSN with SNs monitoring noise, vibration, humidity, temperature, electrical characteristics, and radiation.
- Smart City [46]: A query-based WSN with SNs monitoring motion, location, direction, size, temperature, humidity, radiation, etc.

The contributions of this paper are as follows. First, we develop a model-based analysis methodology to derive a closed form solution of the system lifetime, given operational characteristics of a WSN, and selective capture and smart attack behaviors as input. We consider 3 defense mechanisms to counter selective capture and smart attack: (a) dynamic radio range adjustment; (b) multisource multipath routing for intrusion tolerance; and (c) voting-based intrusion detection. Second, we model smart attacker behaviors and analyze the effect of selective capture on attacker behaviors considering the fact that smart attackers can adjust its attack tactics depending on the malicious node population which is a function of not only time but also distance from the BS because of the presence of selective capture. Lastly, we propose the notion of adaptive network defense management. This involves exploiting the tradeoff between energy consumption vs. reliability gain because of the use of the 3 defenses proposed, and applying the optimal settings dynamically to maximize the lifetime of the WSN incorporating these designs to counter selective capture and smart attacks at runtime.

The rest of the paper is organized as follows. In Section 2, we survey existing work in selective capture and smart attacks, and contrast our work from existing work. In Section 3, we discuss the system model. In Section 4, we discuss the problem definition, and our solution. In Section 5, we propose a model-based analysis methodology to analyze the effects of smart attack and select cap-

H. Al-Hamadi is with the Department of Computer Science, Kuwait University, Khaldiya, Kuwait. Email: hamid@cs.ku.edu.kw. I.R. Chen is with the Department of Computer Science, Virginia Tech, Falls Church, VA 22043. E-mail: irchen@vt.edu.

ture as well as our adaptive network defense management on network dynamics. In Section 6, we conduct a performance analysis. In Section 7 we validate analytical results with extensive simulation using ns3. Finally in Section 8, we conclude the paper and outline future research areas.

2 Related Work

Capture attacks in WSNs can be classified as either random or selective [15], [16], [17]. Selective capture attacks maximize the attack strength by targeting nodes whose capture will result in a high possibility of compromising the basic functionality of the WSN such as data delivery. An intelligent attacker can strategically attack a specific area or a group of sensors to compromise the most number of keys that are not yet compromised [15], [16]. A clever adversary also can strategically attack certain sensors so as to reveal the largest number of unknown pairwise keys [17]. In particular, [17] developed a framework to analyze the effect of selective attacks on performance of key pre-distribution protocols. However, in [15], [16], [17] selective capture was about key compromises and the focus was on key pre-distribution protocol design for achieving resiliency against key compromise attacks.

Our work considers the presence of attackers capable of performing strategic and selective captures of critical SNs near the BS. We note that in the literature, various approaches [1], [18], [42] have been proposed to masquerade and hide critical SNs. In particular, [42] proposed a location-privacy routing protocol to hide the receiver location to counter capture attack. However, energy consumption is generally a concern for SNs in these approaches. Our approach to counter selective capture attack is dynamic redundancy management via multisource multipath routing. We demonstrate the effectiveness of our dynamic redundancy management protocol against selective capture of critical nodes to create black holes near the BS to maximize its attack strength.

We note that range adjustment has been proposed in the literature (e.g., [34-41]) to counter traffic analysis attack and to hide the routing topology. In this paper we use range adjustment to counter selective capture such that a node dynamically adjusts its radio range throughout its lifetime to maintain connectivity with others, as it performs its basic functions of data forwarding (via multipath routing) and intrusion detection (via voting).

An attacker can also employ various smart attack strategies to maximize its attack strength [2]. In [4], attackers do not launch packet dropping attacks to avoid detection. Instead, they inject false data to the data collector. In [19], malicious nodes decrease their attack rate to disguise themselves and avoid being detected by intrusion detection. In [20], [28], in the context of cyber physical systems, the authors considered two attack behaviors: reckless attacks (persistently attacking to impair the system) and random (on-off attacking to avoid intrusion detection). Similar to [20], [28], we consider inside attackers that can perform persistent and random attacks. Furthermore, we consider smart attackers that can perform "opportunistic attacks" (triggered only when opportunities arise), and "insidious attacks" (triggered only when a critical mass of compromised nodes is accumulated). We are the first to consider the effect of selective capture on attack behaviors taking into account the fact that smart attackers very likely will adjust their attack tactics, depending on the malicious node population which is a function of not only time but also distance from the BS because of the presence of selective capture.

In the area of redundancy management of WSNs, the issue of how many paths should be used to maximize the system lifetime was very recently addressed in [21], [22] in the context of multipath routing from a source node to a sink node in a clustered WSN environment. In particular, AFTQC in [22] identifies the best path redundancy to apply to best trade energy consumption for reliability gain to maximize the WSN system lifetime. However, no presence of malicious nodes was considered. Relative to [22], our work considers dynamic redundancy for both fault/intrusion tolerance (via multisource multipath routing) as well as for intrusion detection (via voting) in response to changing environment conditions with the goal to maximize the WSN lifetime. Later [21] enhanced [22] by considering the presence of malicious nodes performing packet dropping attacks. However neither selective capture nor smart attack behavior was considered. Relative to [21], [22], the contribution of this work is to formally analyze the effects of three defenses (one of which is multipath routing) against smart attack and selective capture to maximize the WSN lifetime in a BS-based WSN environment.

This paper extends [29] (which considers smart capture only) by analyzing adaptive network defense management against both smart attack and selective capture. In particular, Section 5 (Analytical Model) is substantially expanded to provide a model-based analysis for random, opportunistic and insidious attacks and its combined effect with selective capture on network dynamics. Section 6 (Performance Evaluation) is substantially extended to include sensitivity analyses on selective capture, smart attack, and adaptive network defense management for countering both selective capture and smart attack. Section 2 (Related Work) is newly created to survey existing work with defense mechanisms against smart attack and selective capture in WSNs and to contrast our approach with existing ones. Section 7 (Simulation Validation) is also newly created to validate analytical results with extensive simulation using ns3.

3 System Model

3.1 WSN Environments

We consider a *query-based* WSN with low-power SNs distributed in a geographic area. There is a base station assigned to the WSN that interconnects the WSN to the outside world and that fields queries from the outside world for sensing results. Queries arrive at the system following a Poisson process with rate λ_q. A query failure is considered as a critical system failure. The initial energy of each SN is E_o^{SN}. Because the WSN application environment considered in this paper is BS-based, SNs are assumed to be deployed in a circular fashion with the BS at the center with radius r^{BS}. We consider random deployment where SNs are deployed randomly (e.g., through air drop) and distributed according to homogeneous spatial Poisson processes with density λ_o^{SN}. The homogeneous spatial Poisson model is frequently used in WSN research [30], [31], [32] since in the absence of extensive statistical studies of spatial node distribution in real WSNs, it provides first-order approximation accounting for stochastic factors in the connectivity process [33]. The expected number of SNs initially in the system thus is $N_o^{SN} = \lambda_o^{SN}\pi(r^{BS})^2$.

3.2 Selective Capture and Smart Attack Model

All SNs are subject to capture attacks. With "selective capture," the adversaries (humans or robots) strategically capture SNs and turn them into *inside* attackers. We represent the capture rate of a SN at a distance x away from the BS at time t by $\lambda_c^{SN}(x,t)$. In practice, one wouldn't know how the capture rate varies as a function of distance to the BS. This requires some prior knowledge or history data analysis. Our analysis methodology developed in the paper is generally applicable as long as the capture rate can be expressed as a function of distance to the BS. Since SNs close to the BS are desirable targets for capture attack, we assume that the capture rates decreases linearly as the SN is further away from the BS as follows:

$$\lambda_c^{SN}(x,t) = \lambda_c^{max} - \frac{x}{r^{BS}}(\lambda_c^{max} - \lambda_c^{min}) \tag{1}$$

where λ_c^{max} is the maximum capture rate the adversary can possibly have, and λ_c^{min} is the minimum capture rate. A baseline case against which this linear selective capture case will be compared is "random capture" by which the adversary, given the same energy and capacity, randomly performs capture attacks, i.e., $\lambda_c^{random} = (\lambda_c^{max} + \lambda_c^{min})/2$ at all distances. We note that these two capture models have the same overall capture rate accounting for the overall capability of the capturers in the system.

After a node is compromised it becomes an inside attacker. An inside attacker can perform packet dropping and data modification attacks [3]. Using our defense mechanism, a data modification attack by a forwarding SN can be detected and the packet will be discarded. So a data modifications attack has the same effect of a packet dropping attack. A malicious node can also perform slandering attacks by recommending a good node as a bad node, and a bad node as a good node when participating in intrusion detection activities. As a result, slandering attacks can cause good nodes being misdiagnosed and evicted from the system, and bad nodes being missed and remained in the system. This effectively creates an area with a high concentration of bad nodes, especially for critical SN areas with a high capture rate under selective capture.

In the literature it is often assumed that an inside attacker performs attacks constantly, without giving consideration to evade intrusion detection. In this paper we characterize a smart attacker by its capability to perform *random*, *opportunistic* and *insidious* attacks. First of all, a smart attacker can perform random or on-off attacks, i.e., attacking with a random probability p_a, to evade intrusion detection. Second, a smart attacker can perform opportunistic attacks, i.e., it attacks only when it sees opportunities which can lead to a successful attack while still eluding detection. Finally, it can perform insidious attacks, i.e., it can perform on-off attacks to evade intrusion detection until a critical mass of compromised node population is reached after which it performs "all in" attacks ($p_a = 1$) to cripple the system totally.

3.3 Defenses against Attacks

Our first defense against selective capture and smart attack is dynamic radio range adjustment. With random deployment, the initial radio range is denoted by r_o^{SN}such that a SN is able to connect to n_0 neighbors for maintaining network connectivity. A SN adjusts its radio range dynamically throughout its lifetime to maintain connectivity such that the average number of 1-hop neighbor SNs remains at $n_{0.}$ Thus, SNs closer to the BS may have to increase radio range more than SNs away from the BS to counter selective capture. Any communication between two nodes with a distance greater than a single hop radio range between them would require multi-hop routing.

Our second defense is multipath routing for intrusion tolerance. This is achieved through two forms of redundancy: (a) source redundancy by which m_s SNs sensing a physical phenomenon in the same feature zone are used to forward sensing data to the BS; (b) path redundancy by which m_p paths are used to relay packets from a source SN to the BS. We assume geographic forwarding is being used to packet routing; thus, no path information is maintained.

While data delivery could fail due to hardware failure and transmission failure because of noise and interference [4], we only consider failure caused by compromised nodes performing packet drop or data modification attacks. We assume that SNs operate in power saving mode (e.g. [5], [6]). Thus, a SN is either active (transmitting or receiving) or in sleep mode. For the transmission and reception energy consumption of sensors, we adopt the energy model in [7] for SNs. We assume that the BS will have pairwise keys with the SNs. A SN also has a pairwise key with each of its neighbors, up to a few hops away for expandability. Thus, a SN can encrypt data for confidentiality and authentication purposes.

Our last defense is voting-based intrusion detection system (IDS) mechanisms to detect and evict compromised nodes. Every SN runs a simple host IDS to assess its neighbors. The host IDS is light-weight to conserve energy. It is also generic and does not rely on the feedback mechanism tied in with a specific routing protocol (e.g., MDMP for WSNS [8] or AODV for MANETs [9]). It is based on local monitoring. That is, each node monitors its neighbor nodes only. Each node uses a set of anomaly detection rules such as a high discrepancy in the sensor reading or recommendation has been experienced, a packet is not forwarded as requested, as well as interval, retransmission, repetition, and delay rules as in [10], [11], [12]. If the count exceeds a system-defined threshold, a neighbor node that is being monitored is considered compromised.

The imperfection of monitoring due to ambient noise and channel error is modeled by a monitoring error probability $p_{err.}$ The host IDS false positive probability (H_{pfp}) for misidentifying a good node as a bad node is due to this monitoring error, so $H_{pfp} = p_{err.}$ On the other hand, the host IDS false negative probability (H_{pfn}) for misidentifying a bad node as a good node is due to this monitoring error as well as how often a bad node performs attacks (with probability p_a). Hence, $H_{pfn} = (1 - p_a) + p_{err}$ upper bounded by 1. A voting-based distributed IDS is applied periodically in every T_{IDS} time interval. A SN is being assessed by its neighbor SNs. In each interval, m neighbor SNs around a target SN will be chosen randomly as voters and cast their votes based on their host IDS results to collectively decide if the target SN is still a good node. The m voters share their votes through secure transmission using their pairwise keys. When the majority of voters come to the conclusion that a target node is bad, then the target node is evicted. There is a system-level false positive probability P_{fp}^{SN} that the voters can incorrectly identify a good node as a bad node. There is also a system-level false negative probability P_{fn}^{SN} that the voters can incorrectly misidentify a bad node as a good node. In the paper, we will derive the two system-level IDS probabilities based on slandering attacks performed by smart attackers exhibiting random, opportunistic and insidious attack behaviors. Finally we note that the system's intrusion detection strength is modeled by the detection interval T_{IDS} (the shorter the stronger) and the number of neighbor voters m (the more the stronger).

3.4 Mean Time to Failure

To provide a unifying metric that considers the above two design tradeoffs, we define the expected number of queries the system can answer correctly until it fails as the expected *lifetime* or the *mean time to failure* (the MTTF) of the system, which can be translated into the actual system lifetime span given the query arrival rate. A failure occurs when no response toward a query is received. The cause could be due to packet dropping or data modification by malicious forwarding SNs, or energy exhaustion. A failure can also occur when the BS receives majority false responses because the majority of *m_s* source SNs selected are malicious.

4 Problem Definition, Solution, and Algorithm Description

The problem we are solving is that given a set of input parameters values characterizing the BS-based WSN operational environment, selective capture and smart attack behaviors as defined in Section 3, we want to dynamically apply the best decision variable settings to maximize the lifetime of the WSN in terms of its MTTF. The decision variables are those defined for the 3 defenses against selective capture and smart attacks, namely, the connectivity degree parameter (n_0) for dynamic radio range adjustment, the path redundancy (m_p) and source redundancy (m_s) for multisource multipath routing, and the number of voters (m) and intrusion invocation interval (T_{IDS}) for voting-based intrusion detection. Our solution methodology is to model network dynamics by means stochastic processes (in Section 5) to yield a closed form solution for the MTTF as a function of these decision variables. The optimal decision variable settings obtained at design time are stored in a table and then applied at runtime by means of O(1) table lookup to implement adaptive network defense management.

We use a flowchart as shown in Figure 1 to describe our adaptive network defense management algorithm. All nodes in the system act periodically to a "T_D timer" event to adjust the optimal parameter setting in response to changing environments. This is indicated by an "event is T_D timer" box. The optimal design settings in terms of optimal $n_0, T_{IDS}, m, m_s,$ and m_p are determined at design time and pre-stored in a table over perceivable ranges of input parameter values. The BS applies a table lookup operation with extrapolation techniques [23-25] to determine the optimal design parameter settings. The complexity is O(1) because of the table lookup technique employed. The action performed by a BS upon a T_D timer event includes determining $n_0, T_{IDS}, m, m_s,$ and m_p based on runtime knowledge of node density and capture rate experienced; and notifying SNs of the new n_0, T_{IDS} and m settings. The action performed by a SN upon this T_D timer event is to adjust its radio range to maintain SN connectivity and to update its n_0, T_{IDS} and m settings. These actions are specified in the two action boxes to the right of the "event is T_D timer" box, with "BS" and "SN" labeling the agents involved.

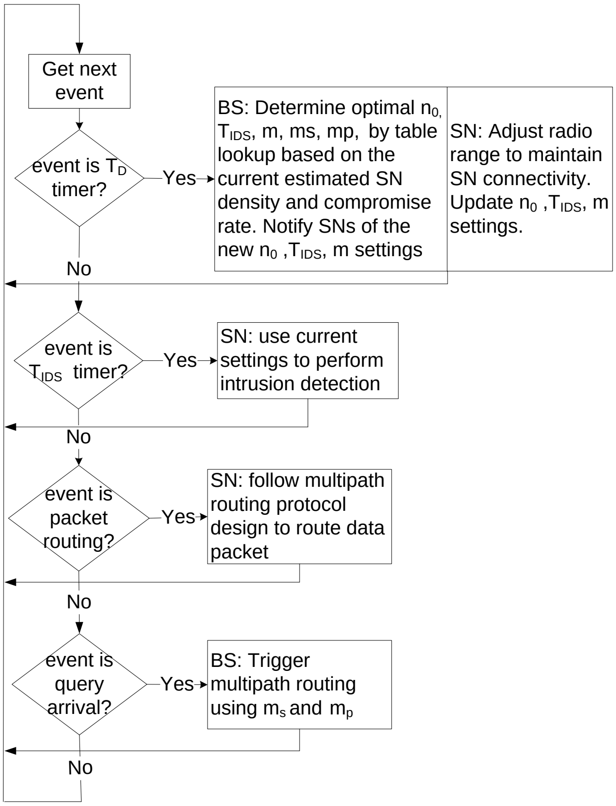

Get next event
event is T_D timer?
Yes
BS: Determine optimal n_0, T_{IDS}, m, ms, mp, by table lookup based on the current estimated SN density and compromise rate. Notify SNs of the new n_0 ,T_{IDS}, m settings
SN: Adjust radio range to maintain SN connectivity. Update n_0 ,T_{IDS}, m settings.
No
event is T_{IDS} timer?
Yes
SN: use current settings to perform intrusion detection
No
event is packet routing?
Yes
SN: follow multipath routing protocol design to route data packet
No
event is query arrival?
Yes
BS: Trigger multipath routing using m_s and m_p
No

Fig. 1: Adaptive Network Defense Management Algorithm Flowchart.

When the T_{IDS} timer event happens (indicated by the "event is T_{IDS} timer" box), each SN uses it current T_{IDS} and m settings to perform intrusion detection. This action is specified in the single action box to the right of the "event is T_{IDS} timer" box, with "SN" labeling the agents involved. When a data packet arrival event occurs (indicated by the "event is packet routing" box), each SN simply follows the prescribed multipath routing protocol to route the packet. This action is specified in the single action box to the right of the "event is packet routing" box, with "SN" labeling the agents involved. When the BS receives a query from a user (indicated by the "event is query arrival" box), it triggers multipath routing for intrusion tolerance using the current optimal (m_s, m_p) setting to prolong the system useful lifetime. The complexity is also O(1) for the BS. This action is specified in the single action box to the right of the "event is query arrival" box, with "BS" labeling the agent involved.

5 Analytical Model

In this section we develop an analytical model to get a close form solution of the MTTF of a query-based WSN built with the three defense mechanisms in the protocol design. The novelty lies in the way we estimate the densities of good nodes, "active" bad nodes, and "inactive" bad node as a function of *location* and *time*, given a set of input parameter values charactering the operational and environmental conditions, and adversary behaviors. This allows us to estimate if a bad node is performing attacks or not at location x and time t, and, consequently, if it will perform packet dropping data modification, and slandering attacks. Consequently, we can reasonably predict if data delivery through a node at location x and time t will succeed or fail.

A model parameter in our formulation can be an *input*, *derived*, *design* or *output* parameter. Specifically, n_0(connectivity degree), m_p (path redundancy), m_s (source redundancy), m (the number of voters for intrusion detection) and T_{IDS} (the intrusion detection interval) are design parameters whose values are to be identified to maximize the system MTTF. Derived parameters are those deriving from input parameters. There is only one output parameter, namely, the MTTF. Note that most derived parameters are dynamic, i.e., as a function of time. For example, SN density, denoted by $\lambda^{SN}(x,t)$, decreases over time because of node failure/eviction as time progresses. On the other hand, radio range, denoted by $r^{SN}(x,t)$, increases over time to maintain connectivity.

The basic idea of our MTTF formulation is that we first deduce the maximum number of queries, N_q, the system can possibly handle before running into energy exhaustion for the best case in which all queries are processed successfully. Because the system evolves dynamically, the amount of energy spent per query also varies dynamically. Given the average interval between query arrivals being $1/\lambda_q$, we can reasonably estimate the amount of energy spent due to query processing and intrusion detection for query j based on the query arrival time $t_{Q,j}$. We then derive the corresponding query success probability $R_q(t_{Q,j})$, that is, the probability that the response to query j arriving at time $t_{Q,j}$ is delivered successfully to the BS. Finally, we compute the MTTF as the probability-weighted average of the number of queries the system can handle without experiencing any failure. More specifically, the MTTF is computed by:

$$MTTF = \sum_{i=1}^{N_q-1} i\left(\prod_{j=1}^{i} R_q(t_{Q,j})\right)\left(1 - R_q(t_{Q,i+1})\right) + N_q \prod_{j=1}^{N_q} R_q(t_{Q,j}) \quad (2)$$

Here $\left(\prod_{j=1}^{i} R_q(t_{Q,j})\right)\left(1 - R_q(t_{Q,i+1})\right)$ accounts for the probability of the system being able to successfully execute i consecutive queries but failing the $i+1$th query. The second term is for the best case in which all queries are processed successfully without experiencing any failure for which the system will have the longest lifetime span.

5.1 Modeling Network Dynamics due to Capture

Let $\lambda^{SN}(x,t)$ represent the density of SNs at distance x from the BS at time t. Initially at deployment time all SNs are good nodes, so $\lambda^{SN}(x,0) = \lambda_0^{SN}$ for all x's.

As time progresses some SNs are captured and turned into compromised nodes. Moreover, some SNs may be diagnosed as bad nodes and get evicted from the system. Let T be the capture time of a SN following a distribution function $F_c(t)$. Then, the probability that a SN at location x away from the BS is compromised at time t, given that it was a good node at time t-T_{IDS}, denoted by $P_c^{SN}(x,t)$, is given by:

$$\begin{aligned} P_c^{SN}(x,t) &= 1 - P\{T > t \mid T > t - T_{IDS}\} \\ &= 1 - \frac{P\{T > t, T > t - T_{IDS}\}}{P\{T > t - T_{IDS}\}} = 1 - \frac{1 - F_c(t)}{1 - F_c(t - T_{IDS})} \end{aligned} \quad (3)$$

In the special case in which the capture time is exponentially distributed, $P_c^{SN}(x,t) = 1 - e^{-\lambda_c^{SN}(x,t) \times T_{IDS}}$ for a SN at distance x from the BS. Recall that the voting-based distributed IDS executes periodically with T_{IDS} being the interval. At the ith IDS execution time (denoted by $t_{I,i}$), a good SN at distance x from the BS may be compromised with probability$P_c^{SN}(x, t_{I,i})$ since the previous IDS execution time $(t_{I,i-1})$. Let $\lambda_{good}^{SN}(x,t)$ and $\lambda_{bad}^{SN}(x,t)$ denote the densities of good and bad SNs at distance x from the BS at time t, respectively. Then, the densities of good and bad SNs at time $t_{I,i}$ just prior to IDS execution can be recursively estimated from the densities of good and bad SNs at time $t_{I,i-1}$ by:

$$\begin{aligned} \lambda_{good}^{SN}(x, t_{I,i}) &= \lambda_{good}^{SN}(x, t_{I,i-1}) \\ &\quad - P_c^{SN}(x, t_{I,i}) \lambda_{good}^{SN}(x, t_{I,i-1}) \\ \lambda_{bad}^{SN}(x, t_{I,i}) &= \lambda_{bad}^{SN}(x, t_{I,i-1}) \\ &\quad + P_c^{SN}(x, t_{I,i}) \lambda_{good}^{SN}(x, t_{I,i-1}) \end{aligned} \quad (4)$$

The boundary conditions are $\lambda_{good}^{SN}(x,0) = \lambda_0^{SN}$ and $\lambda_{bad}^{SN}(x,0) = 0$ for all x's.

As our first defense design, every SN dynamically adjusts its radio range for maintaining connectivity with its peers such that the average number of 1-hop neighbor nodes is n_o to support its intended functions including routing and participating in majority voting IDS as a verifier. In particular, critical SNs (i.e., nodes that are close to the BS) must increase radio range more due to more nodes being evicted as a result of more intensive capture and slandering attacks toward critical SNs.

Let $r^{SN}(x,t)$ denote the radio range of a SN at distance x from its BS at time t so it can find n_o SNs within radio range. Since the SN density is a function of the distance (x) away from the BS, we have to solve $r^{SN}(x,t)$ by integration of the SN population from x-$r^{SN}(x,t)$ to x+$r^{SN}(x,t)$. Let X and Y be two variables denoting the X and Y coordinates in the X-Y coordinate system. Since $Y^2 = r^{SN}(x,t)^2 - X^2$ and $\int_0^{r^{SN}(x,t)} YdX$ gives the area of the upper semicircle, the expected number of SNs covered by radio range, denoted by $r^{SN}(x,t)$, can be obtained by solving the following equation:

$$2\int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda^{SN}(x+X,t)\ \sqrt{r^{SN}(x,t)^2 - X^2}dX = n_o \qquad (5)$$

where the integral gives the expected number of SNs (accounting for density variation along *X*) located in the upper or lower half circle.

For notational convenience, let $n^{SN}(x,t)$ be the average number of neighbor SNs of a SN located at distance x from the BS at time t, $\check{n}^{SN}(x,t)$ be the average number of forwarding neighbors (with f=1/4 for geographical routing), $n_{good}^{SN}(x,t)$ be the average number of good neighbors, and $n_{bad}^{SN}(x,t)$ be the average number of bad neighbors at time t. Since we know the densities of good and bad nodes at time $t_{I,i}$ just prior to IDS execution, we have:

$$n^{SN}(x,t) = 2\int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda^{SN}(t)(x+X,t)\ \sqrt{r^{SN}(x,t)^2 - X^2}dX \qquad (6)$$

$$\check{n}^{SN}(x,t) = \int_{-r^{SN}(x,t)}^{0} \lambda^{SN}(t)(x+X,t)\ \sqrt{r^{SN}(x,t)^2 - X^2}dX$$

$$n_{good}^{SN}(x,t) = 2\int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda_{good}^{SN}(x+X,t)\ \sqrt{r^{SN}(x,t)^2 - X^2}dX$$

$$n_{bad}^{SN}(x,t) = 2\int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda_{bad}^{SN}(x+X,t)\ \sqrt{r^{SN}(x,t)^2 - X^2}dX$$

5.2 Modeling Insidious Attacks

Under insidious attacks, when a malicious node at location x and time t sees more than a threshold percentage, p_{all-in}, of its peers are malicious, it will perform "all-in" attacks. This is modeled by setting p_a=1. Let $P_{bad}(x,t)$ be the percentage of malicious nodes at location x and time t, defined as follows:

$$P_{bad}(x,t) = \frac{\lambda_{bad}^{SN}(x,t)}{\lambda_{bad}^{SN}(x,t) + \lambda_{good}^{SN}(x,t)} \qquad (7)$$

Under the insidious attack, if $P_{bad}(x,t) > p_{all-in}$ then $p_a = 1$. This in turn affects the false positive probability, false negative probability and the probability that a node at location x and time t will perform packet dropping and data modification attacks.

5.3 Modeling Random Attacks

We first derive the false positive probability (P_{fp}^R) and false negative probability (P_{fn}^R) at distance x and time t for this case in which bad nodes perform random attacks with probability $p_{a.}$ Later we extend the derivation to opportunistic attack behavior. Eq. 8 below provides the closed-form solutions for $P_{fp}^R(x,t)$ and $P_{fn}^R(x,t)$ (with x and t omitted for brevity) where n_{bad}^a and n_{bad}^i are the numbers of "active" and "inactive" bad nodes, given by $n_{bad}^{SN}(x,t) \times p_a$ and $n_{bad}^{SN}(x,t) \times (1-p_a)$, respectively; m_{maj} is the minimum majority of m, e.g., 3 is the minimum majority of 5; and ω is H_{pfp} for calculating P_{fp}^R and H_{pfn} for calculating P_{fn}^R. Recall that the imperfection of monitoring due to ambient noise and channel error is modeled by a monitoring error probability p_{err}, so $H_{pfp} = p_{err}$, and $H_{pfn} = (1-p_a) + p_{err}$.

We explain Eq. 8 for the false positive probability P_{fp}^Rat time t below. The explanation to the false negative probability $P_{fn}^R$is similar. A false positive results when the majority of the voters vote against the target node (which is a good node) as compromised. The first term in Eq. 8 accounts for the case in which more than 1/2 of the voters selected from the target node's neighbors are "active" bad nodes who, as a result of actively performing slandering attacks, will always vote a good node as a bad node. Since more than 1/2 of the m voters vote no, the target node (which is a good node) is diagnosed as a bad node in this case, resulting in a false positive. Here the denominator is the total number of combinations to select m voters out of all neighbor nodes, and the numerator is the total number of combinations to select at least m_{maj} bad voters out of n_{bad} nodes and the remaining good voters out of n_{good} nodes.

The second term accounts for the case in which more than 1/2 of the voters selected from the neighbors are good nodes but unfortunately some of these good nodes mistakenly misidentify the target nodes as a bad node with host IDS false positive probability $H_{pfp,}$ resulting in more than 1/2 of the voters (although some of those are good nodes) voting no against the target node. Since more than 1/2 of the m voters vote no, the target node (which is a good node) is also diagnosed as a bad node in this case, again resulting in a false positive. Here the denominator is again the total number of combinations to select m voters out of all neighbor nodes, and the numerator is the total number of combinations to select i "active" bad voters not exceeding the majority m_{maj}, j good or "inactive" bad voters who diagnose incorrectly with $i + j \geq m_{maj}$, and the remaining $m - i - j$ good or "inactive" voters who diagnose correctly. Here we note that an in-

$$P_{fp}^R\ or\ P_{fn}^R =$$

$$\sum_{i=0}^{m-m_{maj}} \left[\frac{C\binom{n_{bad}^a}{m_{maj}+i} \times C\binom{n_{good} + n_{bad}^i}{m-(m_{maj}+i)}}{C\binom{n_{bad}^a + n_{bad}^i + n_{good}}{m}}\right] + \sum_{i=0}^{m-m_{maj}} \left[\frac{C\binom{n_{bad}^a}{i} \times \sum_{j=m_{maj}-i}^{m-i}\left[C\binom{n_{good} + n_{bad}^i}{j} \times \omega^j \times C\binom{n_{good} + n_{bad}^i - j}{m-i-j} \times (1-\omega)^{m-i-j}\right]}{C\binom{n_{bad}^a + n_{bad}^i + n_{good}}{m}}\right] \qquad (8)$$

active "bad" voter acts as if it is a good node to evade detection. Also note that more voters do not necessarily provide better detection accuracy since it depends on the percentage of bad node population. That is, if more bad nodes exist than good nodes in the neighborhood, or good nodes have high host false positive probability (H_{pfp}) and host false negative probability (H_{pfn}), then more voters actually will provide less detection accuracy.

5.4 Modeling Opportunistic Attacks

Under opportunistic attacks, when a malicious voter sees that the majority of voters for intrusion detection of a target node at location x and time t being malicious nodes (active or inactive), it will collude with other malicious nodes and they together (active and inactive) will perform slandering attacks during voting, resulting in $P_{fp} = 1$ and $P_{fn} = 1$. On the other hand, if it does not see the majority being malicious nodes, it will just perform random attacks as usual, resulting inP_{fp}^{R} and P_{fn}^{R}as given in Eq. 8. Summarizing above, the system-level false positive probability (P_{fp}) and false negative probability (P_{fn}) at time t under opportunistic attacks are given by:

$$P_{fp}(x,t) = \begin{cases} 1 & if\ P_{bad}(x,t) \geq\ 0.5 \\ P_{fp}^{R}(x,t) & if\ P_{bad}(x,t) <\ 0.5 \end{cases}$$
$$P_{fn}(x,t) = \begin{cases} 1 & if\ P_{bad}(x,t) \geq\ 0.5 \\ P_{fn}^{R}(x,t) & if\ P_{bad}(x,t) <\ 0.5 \end{cases} \tag{9}$$

5.5 Modeling Network Dynamics due to Intrusion Detection

Our 3rd defense is voting-based IDS to detect and evict suspicious nodes. After the voting-based IDS is executed, however, a good node may be misidentified as a bad node with probability P_{fp} (Eq. 9) and mistakenly removed from the WSN. On the other hand, a bad node may be missed with probability P_{fn}(Eq. 9) and remained in the system. Consequently, we need to adjust the population of good and bad nodes after IDS execution. Let $\overline{\lambda_{good}^{SN}(x,t_{I,\imath})}$ and $\overline{\lambda_{bad}^{SN}(x,t_{I,\imath})}$ denote the densities of good and bad SN nodes located at distance x from the BS, respectively, after IDS execution at time t. Then:

$$\overline{\lambda_{good}^{SN}(x,t_{I,\imath})} = \lambda_{good}^{SN}(x,t_{I,i}) - \lambda_{good}^{SN}(x,t_{I,i}) \times P_{fp}(x,t_{I,i})$$
$$\overline{\lambda_{bad}^{SN}(x,t_{I,\imath})} = \lambda_{bad}^{SN}(x,t_{I,i}) - \lambda_{bad}^{SN}(x,t_{I,i}) \times (1 - P_{fn}(x,t_{I,i})) \tag{10}$$

Therefore, the probability that node j located at distance x from its BS is an "active" bad SN performing packet dropping attacks at time $t_{I,i}$,denoted by $Q_{c,j}^{SN}\ (x,t_{I,i})$, is given by:

$$Q_{c,j}^{SN}\ (x,t_{I,i}) = \frac{\overline{\lambda_{bad}^{SN}(x,t_{I,\imath})}}{\overline{\lambda_{bad}^{SN}(x,t_{I,\imath})} + \overline{\lambda_{good}^{SN}(x,t_{I,\imath})}} \times p_a \tag{11}$$

The first term on the right hand side is the probability node j located at x is a bad node, and the 2nd term is the probability that it is performing attacks (p_a). $Q_{c,j}$ derived above provides critical information because an "active" bad node can perform packet dropping and data modification attacks causing a path to be broken if it is on a path from source SNs to the BS.

We note that the good/bad node density will remain the same until the next IDS execution (after T_{IDS} seconds) because the IDS only detects and evicts nodes periodically (given that typically node hardware/software failure happens less frequently than security failure). The remaining nodes are good nodes that pass the IDS evaluation and bad nodes that are undetected by the IDS. Thus, $\overline{\lambda_{good}^{SN}(x,t_{I,\imath-1})}$and $\overline{\lambda_{bad}^{SN}(x,t_{I,\imath-1})}$ obtained at time $t_{I,i-1}$ essentially become $\lambda_{good}^{SN}(x,t_{I,i}-1)$ and $\lambda_{bad}^{SN}(x,t_{I,i-1})$, respectively, for the next round of IDS execution at time $t_{I,i}$.

We can also estimate the average number of SNs in the WSN at time t as:

$$N^{SN}(t) = \int_{0}^{r^{BS}} \lambda^{SN}(x,t)\ \ 2\pi x\ dx \tag{12}$$

5.6 Query Success Probability

We will use the notation SNj to refer to SN j responsible to relay the packet for the jth hop from the source SN to the BS. Also we will use the notation $x(j)$ to refer to the distance from SNj to its BS.

Let $D_{SN\text{-}BS}$ be the distance between a SN (selected to report sensor readings) and its BS, which on average is $r^{BS}/2$. Then the average numbers of hops to forward data from a source SN to the BS, denoted by N_{hop}^{SN}, can be estimated as follows:

$$\sum_{j=1}^{N_{hop}^{SN}} r^{SN}(x(j),t) = D_{SN-BS} \tag{13}$$

The equation above equates the sum of hop distances with the source-destination distance.

The success probability for SNj to transmit a packet to at least p next-hop SN neighbors (with indices k=1, 2, … p) along the direction of the destination node based on *geographical routing* is given by:

$$\theta_j^{SN}(p) = \sum_{i=p}^{\check{n}^{SN}(x(j),t)} [\binom{\check{n}^{SN}(x(j),t)}{i} \prod_{k=1}^{i}(1 - Q_{k,c}^{SN}(x(k),t)) \prod_{k=i+1}^{\check{n}^{SN}(x(j),t)} Q_{k,c}^{SN}(x(k),t)] \tag{14}$$

where $Q_{k,c}^{SN}(x(k),t)$ is the probability that SNk is compromised as derived in Eq. 11, and $\check{n}^{SN}(x(j),t)$ is the average number of forwarding neighbor SNs for SNj as derived from Eq. 6.

A path starting at SNj to the BS is successful if in each hop there is at least one healthy next-hop SN neighbor found. Thus, the success probability of a path starting from SNj (a source node has index j=1) to the BS is given by:

$$\varphi_j^{SN} = \prod_{l=j}^{N_{hop}^{SN}-1} \theta_l^{SN}(1) \tag{15}$$

Our 2nd defense is to create m_p paths between a source SN and the BS for *path redundancy*. The m_p paths are formed by choosing m_p SNs in the first hop and then choosing only one SN in each of the subsequent hops. The source SN will fail to deliver data to the SN if one of the following happens: (a) none of the SNs in the first hop receives the message; (b) in the first hop, i *(1≤ i<m_p)* SNs receive the message, and each of them attempts to form a path for data delivery; however, all *i* paths fail to deliver the message because the subsequent hops fail to receive the broadcast message; or (c) in the first hop, at least m_p SNs receive the message from the source SN from which m_p SNs are randomly selected to forward data, but all m_p paths fail to deliver the message because the subsequent hops fail to receive the message. Summarizing above, the probability of a source SN (with index *j*=1) failing to deliver data to the BS through multipath routing is given by:

$$Q_1^{SN}(p) = 1 - \theta_1^{SN}(1) + \sum_{p=1}^{m_p} \theta_1^{SN}(p)\,[1 - \varphi_2^{SN}]^p \tag{16}$$

Consequently, the failure probability of data delivery to the BS from m_s source SNs, each utilizing m_p paths to relay data, is given by:

$$Q_f = \left[1 - (1 - Q_1^{SN})\left(1 - Q_{c,1(x(1),t)}^{SN}\right)\right]^{m_s} \tag{17}$$

Therefore, the query success probability is given by:

$$R_q = 1 - Q_f \tag{18}$$

Note that in the above derivation we omit time for brevity. More precisely, R_q derived above should be $R_q(t_{Q,i})$ since the query success probability is a function of time, depending on the node count and population density at the i^{th} query's execution time (i.e., at time $t_{Q,i}$).

5.7 Energy Consumption

Now we estimate the amounts of energy spent by a SN located at distance x away from the BS during a query interval $[t_{Q,i},\ t_{Q,i+1}]$ and an IDS interval $[t_{I,i},\ t_{I,i+1}]$ so as to estimate N_q, the maximum number of queries this SN can possible handle before running into energy exhaustion. When all SNs at distance x consumes all their energy, a 'black ring' at distance x away from the BS is formed. Nodes at distance greater than x will have to increase their radio range in order to maintain connectivity with the BS but eventually the system ceases to function. When selective capture is in effect, one can see that a black ring can more easily develop for nodes close to the BS. To normalize energy consumption over N_q queries, let α be the ratio of the IDS execution rate to the query arrival rate so that αN_q is the numbers of IDS cycles before SN energy exhaustion. Then, we can estimate N_q by the fact that the SN energy consumed due to intrusion detection, and query processing is equal to the initial SN energy as follows:

$$E_o^{SN} = \sum_{i=1}^{N_q} E_q^{SN}(x, t_{Q,i}) + \sum_{i=1}^{\alpha N_q} E_{IDS}^{SN}(x, t_{I,i}) \tag{19}$$

Below we outline how to calculate $E_q^{SN}(x, t_{Q,i})$ and $E_{IDS}^{SN}(x, t_{I,i})$. We first estimate energy consumed by transmission and reception over wireless link. The energy spent by a SN to transmit an encrypted data packet of length n_b bits over a distance *r* is estimated as [7]:

$$E_T^{SN}(r) = n_b\left(E_{elec} + E_{amp} r^z\right) \tag{20}$$

Here E_{elec} is the energy dissipated to run the transmitter and receiver circuitry, E_{amp} is the energy used by the transmit amplifier, and *r* is the transmission radio range. We use the current SN radio range to derive E_T^{SN}. We set E_{amp}= 10 pJ/bit/m^2 and z = 2 when the radio range is less than a threshold distance d_0 (75m) and E_{amp}= 0.0013 pJ/bit/m^4 and z = 4 otherwise [7]. The energy spent by a node to receive an encrypted message of length n_b bits is given by:

$$E_R^{SN} = n_b E_{elec} \tag{21}$$

The energy consumed by a SN located at x for processing the i^{th} query, $E_q^{SN}(x, t_{Q,i})$, conditioning on it is being a data delivery path with probability $P_q^{SN}(x, t_{Q,i})$, is the energy consumed for reception (except when it is a source SN) and transmission, i.e.,

$$E_q^{SN}(x, t_{Q,i}) = P_q^{SN}(x, t_{Q,i}) \times [E_R^{SN} + E_T^{SN}(r^{SN}(x, t_{Q,i}))] \tag{22}$$

Since source SNs are randomly picked to answer a query, the probability that a SN at distance x away from the BS is on the data path $P_q^{SN}(x, t_{Q,i})$ is estimated as the probability of a SN at x is needed for data delivery, $(r^{BS} - x)/r^{BS}$, multiplied with the probability that this particular sensor is needed, $m_p m_s / N^{SN}(x, t_{Q,i})$. $N^{SN}(x, t_{Q,i}) = \int_{x - r^{SN}(x, t_{Q,i})}^{x + r^{SN}(x, t_{Q,i})} \lambda^{SN}(X, t)\ 2\pi X\, dX$ is the expected number of SNs within the radio range of SNs at distance x.

For intrusion detection every node is evaluated by m voters in an IDS cycle, and each voter sends its vote to the other m - 1 voters. Hence, the energy spent by a SN located at x in the i^{th} IDS cycle, $E_{IDS}^{SN}(x, t_{I,i})$, conditioning on it serving as a voter with probability $P_{IDS}^{SN}(x, t_{I,i})$ for each of its $n^{SN}(x, t_{I,i})$ neighbors, is the energy consumed for reception of m-1 votes and transmission of its vote to other m-1 voters, i.e.,

$$E_{IDS}^{SN}(x, t_{I,i}) = P_{IDS}^{SN}(x, t_{I,i}) \times n^{SN}(x, t_{I,i}) \times (m - 1)[E_R + E_T^{SN}\left(r^{SN}(x, t_{Q,i})\right)] \tag{23}$$

Here the probability that a SN at distance x serves as a voter for a neighbor SN, $P_q^{SN}(x, t_{Q,i})$, is estimated as $m/n^{SN}(x, t_{I,i})$.

The system fails when a SN at distance $x = r_{max}^{SN}$ (SN maximum radio range) depletes its energy since there is no way to maintain connectivity even by dynamic range adjustment. That is, we set $x = r_{max}^{SN}$to obtain $E_q(t_{Q,i})$, and $E_{IDS}(t_{I,i})$ from Eqs. 22, and 23, respectively, and then we calculate N_q from Eq. 19. The knowledge of N_q along with $R_q(t_{Q,i})$ in Eq. 18 allows us to calculate the system MTTF given by Eq. 2.

6 PERFORMANCE EVALUATION

In this section, we present numerical results. Our reference WSN consists of N_o^{SN}= 1500 SN nodes initially deployed with density $\lambda_o^{SN}$with the BS sitting at the center of a circular area with radius r^{SN}=300m. The selective SN capture time is assumed to be exponentially distributed following the linear model described by Eq. 1, with λ_c^{min} being once per 4 weeks and λ_c^{max} varying in the range of once per half day to once per 2 weeks. The radio range r_{SN} is dynamically adjusted to maintain network connectivity of n_0= 7 to support basic multipath routing and voting-based IDS functions. The initial energy level of a SN is E_0^{SN} = 2 Joule. The energy parameters used by the radio module are adopted from [7, 14]. The energy dissipation E_{elec} to run the transmitter and receiver circuitry is 50nJ/bit. The energy used by the transmit amplifier to achieve an acceptable signal to noise ratio (E_{amp}) is 10 pJ/bit/m^2 for transmitted distances less than the threshold distance d_0 (75m) and 0.0013 pJ/bit/m^4 otherwise. The query arrival rate λ_q is a variable ranging from 10^{-2} to 1 query/sec to reveal points of interest. The imperfection of host IDS monitoring due to ambient noise and channel error is modeled by a monitoring error probability p_{err}= 1%.

6.1 Analyzing the Effect of Selective Capture on MTTF

We first analyze the effects of selective capture on MTTF. To isolate out the effect of smart attack (which we will consider later), we only consider persistent attackers that always attack with probability 1 (p_a = 1.0). Our objective is to identify the best protocol setting of our defenses against selective capture which converts good nodes into bad nodes. This includes the radio range to be adjusted dynamically by individual SNs, the best redundancy level used for multipath routing, as well as the best redundancy level in terms of the number of voters and the best intrusion invocation interval used for intrusion detection to maximize the WSN lifetime. We use random capture as a baseline case for performance comparison.

6.1.1 Identifying Protocol Optimal Settings

We first examine the effect of the compromise rate and intrusion detection interval T_{IDS} on the optimal (m_p, m_s) to maximize the WSN lifetime. Tables 1 and 2 summarize the optimal (m_p, m_s) values to maximize the WSN lifetime under selective capture and random capture attacks, respectively, with m fixed at 3 (i.e., the number of voters is 3). The maximum compromise rate λ_c^{max} on the 1st column specifies the magnitude of the compromise rate (with λ_c^{min} being fixed at once per 4 weeks). The intrusion detection interval T_{IDS} on the 1st row specifies the IDS interval. For example, when λ_c^{max} =1/(0.5 day) and T_{IDS}=4hrs, the optimal (m_p, m_s) is (2, 5) under selective capture in Table 1 and is (1, 5) under random capture in Table 2.

We first observe that there exists an optimal (m_p, m_s) setting under which the MTTF is maximized for either case. Furthermore, a higher (m_p, m_s) is needed when the capture strength λ_c^{max} increases. Also under selective capture attacks, the system must use a higher redundancy level to maximize the MTTF. For example when T_{IDS} = 4 hrs and λ_c^{max}= 1/(0.5 day), the optimal (m_p, m_s) setting is (2, 5) under selective capture (in Table 1) but is only (1, 5) under random capture attacks (in Table 2). This is because selective capture requires the system to apply more redundancy to cope with more critical nodes being compromised. The system is better off in this case to use higher redundancy to ensure secure routing at the expense of more energy consumption to maximize the system MTTF. We also observe from Tables 1 and 2 that the optimal MTTF is more likely to be achieved using a redundancy setting with a high m_s as opposed to a high m_p. While the same number of total paths can be achieved using various (m_p, m_s) combinations, e.g., 6 paths can be achieved by (1, 6), (2, 3), (3, 2) or (6, 1), increasing m_s rather than increasing m_p can more effectively increase the query success probability because the failure of a single source SN results in a system failure, even if the source SN is connected to the BS via m_p paths. On the other hand, the failure of a single path is less damaging to query success. Furthermore, we expect little difference in terms of energy consumption when the number of paths is the same. As a result, the optimal (m_p, m_s) setting favors a high m_s over a high m_p whenever possible.

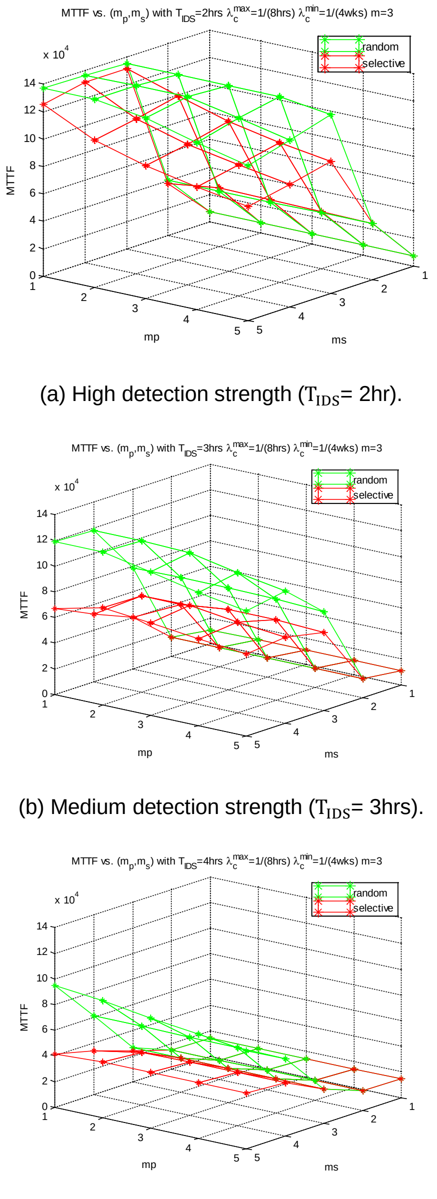
We next analyze the effect of the intrusion detection interval T_{IDS} (representing the intrusion detection strength) on the system MTTF. Whether to use a small or large T_{IDS} depends on the capture strength λ_c^{max}. When the capture strength is high (i.e., when λ_c^{max} is high), as evidenced by the frequency at which bad nodes are detected by the IDS and evicted, we must counter it with high detection strength (a small T_{IDS}). Conversely, when the capture strength is low, a large T_{IDS}could be used to save energy to maximize the MTTF.

TABLE 1: OPTIMAL $(\boldsymbol{m_p}, \boldsymbol{m_s})$ UNDER SELECTIVE CAPTURE WITH VARYING $\boldsymbol{\lambda_c^{max}}$ AND $\boldsymbol{T_{IDS}}$ VALUES.

λ_c^{max}	T_{IDS}=1hr	2hrs	4hrs	6hrs	8hrs
1/(8 hrs)	(1,3)	(1,4)	(5,5)	(5,5)	(5,5)
1/(0.5 day)	(1,2)	(1,4)	(2,5)	(2,5)	(2,5)
1/(0.75 days)	(1,2)	(1,2)	(1,4)	(2,5)	(2,5)
1/day	(1,2)	(1,2)	(1,2)	(1,4)	(2,5)
1/(2 days)	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)

TABLE 2: OPTIMAL $(\boldsymbol{m_p}, \boldsymbol{m_s})$ UNDER RANDOM CAPTURE WITH VARYING $\boldsymbol{\lambda_c^{max}}$ AND $\boldsymbol{T_{IDS}}$ VALUES.

λ_c^{max}	T_{IDS}=1hr	2hrs	4hrs	6hrs	8hrs
1/(8 hrs)	(1,3)	(1,4)	(1,5)	(5,5)	(5,5)
1/(0.5 day)	(1,2)	(1,2)	(1,5)	(1,5)	(2,5)
1/(0.75 days)	(1,2)	(1,2)	(1,2)	(1,5)	(1,5)
1/day	(1,2)	(1,2)	(1,2)	(1,2)	(1,5)
1/(2 days)	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)



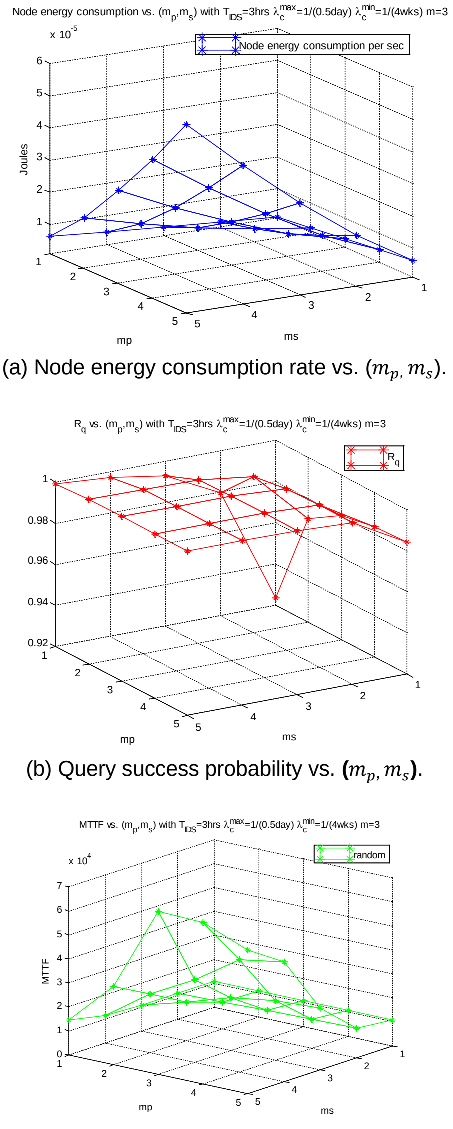
(a) High detection strength (T_{IDS}= 2hr).

(b) Medium detection strength (T_{IDS}= 3hrs).

(c) Low detection strength (T_{IDS}= 4hrs).

Fig. 2: MTTF vs. (m_p, m_s) with varying Detection Strength in the Presence of High Capture Strength.

Figures 2(a), 2(b), and 2(c) show MTTF vs. (m_p, m_s) under small (T_{IDS}=2 hrs), medium (T_{IDS}=3 hrs), and large (T_{IDS}=4 hrs) detection intervals, respectively, for the case when the capture strength is high, i.e., $\lambda_c^{max} = 1/(8\text{ hrs})$. We again set m=3 to isolate its effect. We observe that at the optimal (m_p, m_s) setting, the MTTF under T_{IDS}=2 hrs (Figure 2(a)) is much higher than the MTTF under T_{IDS}=4 hrs (Figure 2(c)). This is because when the system is subject to a high capture rate, the system is better off to apply high detection strength (a small T_{IDS} at 2 hrs) at the expense of more energy consumption to quickly detect and evict compromised nodes, instead of applying low detection strength (a large T_{IDS} at 4 hrs), so as to increase the MTTF. This trend applies to both selective and random capture attacks. We also see that the MTTF under selective capture is much lower than that under random capture because with selective capture critical nodes are more easily compromised and black holes can more easily form near the BS to cause a system failure. Lastly we note that optimal (m_p, m_s) setting is highly situation dependent. The optimal (m_p, m_s) settings under selective capture in Figures 2(a), 2(b) and 2(c) are (1, 4), (2, 5), and (5, 5) respectively.



(a) Node energy consumption rate vs. (m_p, m_s).

(b) Query success probability vs. (m_p, m_s).

(c) MTTF vs. (m_p, m_s).

Fig. 3: An Example showing (a) Node Energy Consumption per second, (b) Query Success Probability, and (c) MTTF vs. (m_p, m_s).

6.1.2 *Tradeoff Analysis between Energy and Reliability*

The optimal (m_p, m_s) setting identified to maximize MTTF is a result of the tradeoff between query success probability (Eq. 18) vs. energy consumption rate (Eq. 19). This is illustrated in Figure 3 for a selective capture case where Figure 3(a) shows the average node energy consumption rate vs. (m_p, m_s) (for both query processing and IDS execution), Figure 3(b) correspondingly shows the query success probability (R_q) vs. (m_p, m_s), and Figure 3(c) shows the MTTF vs. (m_p, m_s) as a result of this tradeoff. Here we take data collected over the lifetime for Figures 3(a) and 3(b), since the first two quantities are time dependent. While the query success probability is maximized with (m_p, m_s)=(5, 5), this setting consumes the most energy which adversely shortens the system lifetime. As a result, the optimal (m_p, m_s) setting that maximizes MTTF in this example is (3, 5).

6.2 Analyzing the Effect of Smart Attack on MTTF

In this section, we analyze the effect of smart attack on MTTF. We consider 3 smart attack behavior models: (1) random, (2) random + opportunistic, and (3) random + opportunistic + insidious. We use persistent attacks as the baseline case for performance comparison. Also for each case, we differentiate selective capture from random capture as smart attackers will exploit environment vulnerability due to selective capture which creates more malicious nodes in areas nearer to the BS.

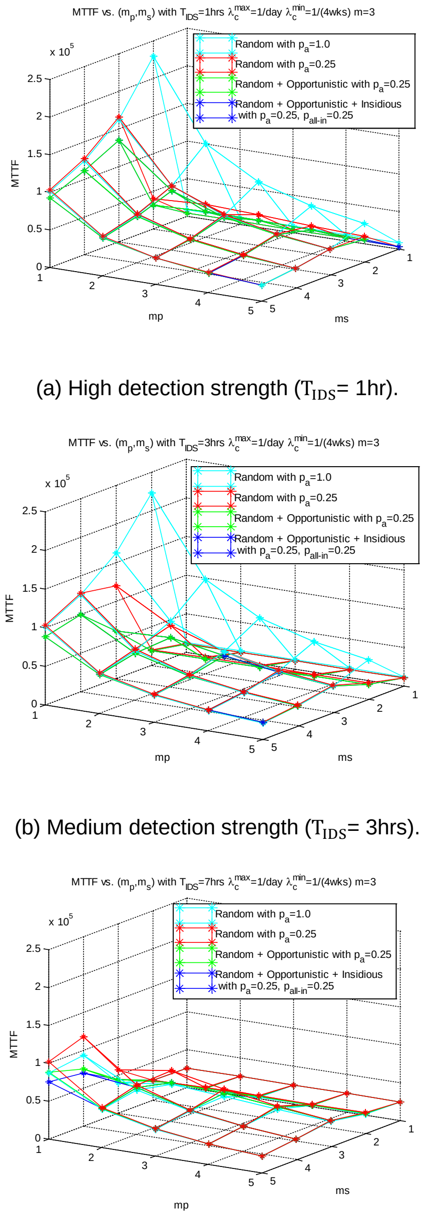
6.2.1 Smart Attack under Random Capture

Figures 4(a)-(c) show MTTF vs. (m_p, m_s) for a WSN under smart attack and random capture when the detection strength goes from high to low (T_{IDS}= 1hr, 3hrs to 7hrs). For random attacks, $p_a = 0.25$. For insidious attacks, $p_{all-in} = 0.25$.

There are several general observations. First of all, we observe that there exists an optimal (m_p, m_s) setting under which the MTTF is maximized for each attack behavior model. Further, more damaging attacks in general require higher redundancy to achieve the optimal (m_p, m_s) setting. For example, in Figure 4(a), the optimal (m_p, m_s) settings for the persistent attack model ($P_a = 1.0$) and the random + opportunistic + insidious attack model are (1,2) and (1,3), respectively.

Second, from the smart attacker's perspective, the best attack strategy depends on the defender's detection strength. We observe that when the detection strength is high (Figure 4(a)) the best adversary strategy is to attack randomly with a low random attack probability so as to evade detection and wait opportunistically for the best chance to come by to attack. This is observed in Figure 4(a) where we see that in this case attacking persistently with $p_a = 1$ (the top curve) results in the attackers being detected and removed from the system, and consequently yields the highest MTTF among all cases. Conversely, when the detection strength is low (Figure 4(c)) the best adversary strategy is to attack with a high random attack probability to maximize the damage. This is observe in Figure 4(c) where we see that attacking with a low random attack probability $p_a = 0.25$ (the top curve) actually results in the highest MTTF among all, since in this case intrusion detection is ineffective; thus, attacking persistently (with $p_a = 1$) is more effective than random attacks with a low random attack probability $p_a = 0.25$.

Lastly, we observe that the effect of insidious attacks is pronounced when the detection strength is low. In Figure 4(a) where the detection strength is high, its effect is insignificant. This is evidenced in Figure 4(a) that MTTF under the random + opportunistic + insidious attack model is almost the same as that under the random + opportunistic attack model (the bottom curve). However, when the detection strength decreases, the effect becomes manifested because of a much higher chance of many malicious nodes accumulated in the system due to weak detection. This is evidenced in Figure 4(c) where MTTF under the random + opportunistic + insidious attack model (bottom curve) is much lower than that under the random + opportunistic attack model (3rd bottom curve).



(a) High detection strength (T_{IDS}= 1hr).

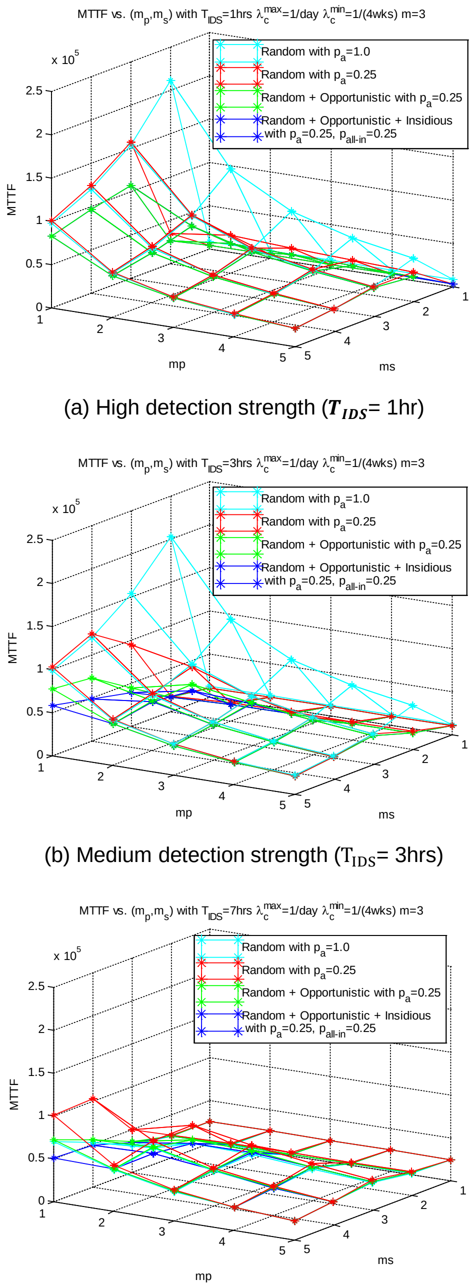
(b) Medium detection strength (T_{IDS}= 3hrs).

(c) Low detection strength (T_{IDS}= 7hrs).

Fig. 4: MTTF vs. (m_p, m_s) under Smart Attack and Random Capture.

6.2.2 Smart Attack under Selective Capture

We repeat the same analysis as in the last section above except that we analyze the effect of smart attack on MTTF of a WSN with selective capture. We summarize the results by Figures 5(a)-(c) showing MTTF vs. (m_p, m_s) under smart attack and selective capture, with the detection strength going from high to low (T_{IDS}= 1hr, 3hrs to 7hrs), respectively.



(a) High detection strength ($\boldsymbol{T_{IDS}}$= 1hr)

(b) Medium detection strength (T_{IDS}= 3hrs)

(c) Low detection strength (T_{IDS}= 7hrs)

Fig. 5: MTTF vs. (m_p, m_s) under Smart Attack and Selective Capture.

Comparing Figures 4 and 5, we see that the same general observations drawn from Figure 4 apply. However, there are two striking differences. First, the MTTF value obtained, given the same detection strength and attack behavior model, is considerably lower than that under random capture. This illustrates the damaging effect of selective capture which creates black holes near the BS to maximize its attack strength. Second, as the intrusion detection strength decreases, the difference in MTTF between random capture and selective capture widens (e.g., Figure 4(c) vs. Figure 5(c)) because of the damaging effect of selective capture which causes more malicious nodes accumulated near the BS.

6.3 Countering Selective Capture and Smart Attack

In this section, we analyze the effectiveness of our defenses against selective capture and random + opportunistic + insidious attacks which we model by λ_c^{max}, λ_c^{min}, p_a, and p_{all-in}. Figure 6 shows the optimal T_{IDS} interval (representing detection strength) to counter such attackers with varying capture strengths (λ_c^{max}) and random attack probability (p_a). In Figure 6, we fix λ_c^{min} to once per 4 weeks and vary λ_c^{max} in the range of once per half day to once per 2 weeks. We also fix $p_{all-in} = 0.25$ to isolate its effect. We observe from Figure 6 that a low $p_a$demands a high detection rate (i.e., a small T_{IDS} interval). The reason is that a low $p_a$will result in a high per-host false negative probabilityH_{pfn}. Consequently, to cope with many hidden bad nodes missed by intrusion detection, the system will have to use a small T_{IDS} interval for high detection strength. Another observation is that as λ_c^{max}increases, T_{IDS}decreases (or the detection strength increases) to counter the increasing capture rate.

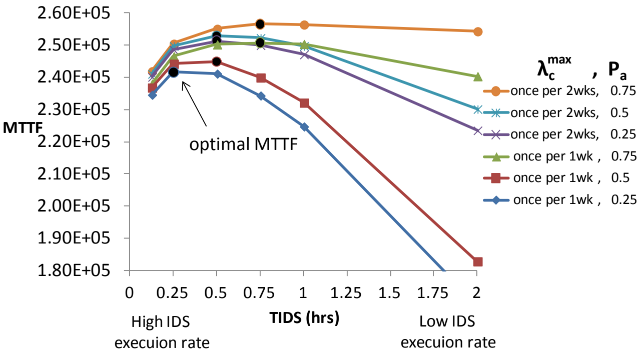


Fig. 6: Countering Selective Capture and Random + Opportunistic + Insidious Attacks with varying $\boldsymbol{\lambda_c^{max}}$ and $\boldsymbol{p_a}$ by Adjusting Detection Strength Parameter $\boldsymbol{T_{IDS}}$.

TABLE 3: OPTIMAL (m_p, m_s) UNDER LOW $P_a(0.25)$, WITH VARYING λ_c^{max}AND T_{IDS}.

	T_{IDS}= 0.25hr	0.5hrs	1hr	2hrs	4hrs
λ_c^{max} = 1/(0.5 day)	(1,3)	(1,4)	(2,4)	(2,5)	(3,5)
λ_c^{max}= 1/(1 day)	(1,3)	(1,3)	(1,3)	(1,5)	(2,4)
λ_c^{max}= 1/(7 days)	(1,2)	(1,2)	(1,2)	(1,3)	(1,3)
λ_c^{max}= 1/(14 days)	(1,2)	(1,2)	(1,2)	(1,2)	(1,3)

TABLE 4: OPTIMAL (m_p, m_s) UNDER HIGH $P_a(0.75)$, WITH VARYING λ_c^{max}AND T_{IDS}.

	T_{IDS}= 0.25hr	0.5hr	1hr	2hrs	4hrs
λ_c^{max}= 1/(0.5 day)	(1,3)	(1,3)	(1,3)	(1,5)	(2,5)
λ_c^{max}= 1/(1 day)	(1,2)	(1,2)	(1,3)	(1,3)	(1,5)
λ_c^{max}= 1/(7 days)	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)
λ_c^{max}= 1/(14 days)	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)

Tables 3 and 4 summarize the optimal $(m_{p,}m_s)$ under low and high p_a values, respectively, when there are selective capture and random + opportunistic + insidious attacks. We observe that the more hidden the inside attackers are, that is, as $p_a$decreases, the more ($m_{p,}$ m_s) redundancy is required to cope with the bad node population accumulated due to misses in intrusion detection. This is evidenced by comparing optimal $(m_{p,}m_s)$ settings listed in Tables 3 and 4 for the same $\lambda_c^{max}$and T_{IDS}.

Next we analyze the effect of p_{all-in} (the all-in attack percentage population threshold) on MTTF of a WSN subject to selective capture with random + opportunistic + insidious attacks. We first note that a small p_{all-in} means that malicious nodes will perform all-in attacks early on (i.e., setting p_a=1) as soon as it senses the small percentage population threshold is reached. On the other hand, a large p_{all-in}means that malicious nodes will stay hidden until the large percentage population threshold is reached.

Table 5 summarizes this trend with $p_a = 0.1$ over a wide range of λ_c^{max}values. This seemingly odd trend has a logical explanation. That is, when $p_{all-in}$is very small (say 0-10%), this critical mass of malicious nodes can be reached early on after which p_a becomes 1 (jumping from 0.1) for all-in attacks. So malicious nodes can be easily detected and the system should just use moderate detection strength to balance energy consumption with detection rate. As p_{all-in}increases further (say 10-20%), malicious nodes stay hidden until the all-in attack percentage population threshold is reached, so the system should use high detection strength to remove malicious nodes to prevent this critical mass from reached. Finally as p_{all-in}continues to increase past a high threshold (say >25%), insidious attacks will be ineffective since it is unlikely such a high critical mass can be reached, given that $p_a = 0.1$ so a bad node performing random attacks can still be detected by the IDS. The system in this case is better off using just moderate detection strength to balance the energy consumption rate with the IDS detection rate.

TABLE 5: OPTIMAL T_{IDS} UNDER VARYING λ_c^{max}AND p_{all-in} (WITH $p_a = 0.1$).

	p_{all-in} =0.0	0.1	0.15	0.2	0.25
λ_c^{max}= 1/(0.5 day)	0.75	0.75	0.25	0.125	0.5
λ_c^{max}= 1/(1 day)	1	1	0.5	0.25	0.5
λ_c^{max}= 1/(2 days)	2	2	0.75	0.25	1
λ_c^{max}= 1/(3 days)	3	3	0.5	0.25	3

TABLE 6: OPTIMAL $(m_{p,} m_s)$ UNDER VARYING λ_c^{max}AND p_{all-in} (WITH $T_{IDS} = 0.5$ AND $p_a = 0.1$).

	p_{all-in} =0.0	0.1	0.2	0.3
λ_c^{max}=1/(0.5 day)	(1,2)	(1,3)	(1,3)	(2,4)
λ_c^{max}= 1/(1 day)	(1,2)	(1,2)	(1,3)	(1,5)
λ_c^{max}= 1/(2 days)	(1,2)	(1,2)	(1,2)	(2,2)
λ_c^{max}= 1/(3 days)	(1,2)	(1,2)	(1,2)	(2,2)

Finally, Table 6 summarizes the optimal $(m_{p,}m_s)$ settings (representing multipath multisource redundancy for intrusion tolerance) to maximize MTTF obtainable under varying p_{all-in}settings. We fix T_{IDS} to 0.5 hrs to isolate its effect. We can see from Table VI that as p_{all-in} increases attackers become more hidden, and, consequently, a higher level of ($m_{p,}$ m_s) redundancy is required to cope with the bad node population accumulated due to misses in intrusion detection. This finding is consistent with that drawn from Tables 3 and 4.

7 SIMULATION VALIDATION

In this section, we present simulation results to compare with analytical results for the purpose of validation. We use the ns-3 discrete-event network simulator as our simulation framework. The ns-3 implementation of a BS-based WSN closely follows our analytical model in Section 6 except that we allow backtracking, which is used in practice for geographic routing but is not modeled in our analytical model due to complexity. Specifically, in case a forwarding node nearer to the BS cannot be found, the packet may be routed to a node that is farther away from the BS, in the hope that a path will eventually be found to reach the BS. We limit backtracking to go backward for a maximum of 3 hops after which the path is considered failed.

Below we report simulation results. Each data point of MTTF is generated from 100 simulation runs. In each run, an observation of MTTF is collected when a system failure condition defined in Section 3.4 is satisfied.

Figures 7 and 8 compare simulation results with analytical results for MTTF vs. $(m_{p,}\ m_s)$ under random and selective capture attacks, respectively. We see that there is a remarkable match between simulation and analytical results, both showing the same optimal $(m_{p,}\ m_s)$ under which MTTF is maximized. The MTTF values generated from the analytical model are consistently higher than those generated from simulation. We attribute this discrepancy to two reasons. The first reason is that a path in simulation in general has a longer overall length than that in the analytical model which assumes a straight line distance from the source SN to the BS. Consequently, there are more intermediate nodes on a path and the path failure probability is higher. The second reason is due to the assumption of disjoint paths in the analytical model. This assumption in general is justified when there are sufficient nodes. However as nodes are evicted because of intrusion detection, it may not be justified. While the analytical model continues to use the disjoint path assumption for MTTF calculation, the simulation uses backtracking to cope with a lack of SNs for forming m_p paths, especially when the frontier of a path is close to the BS which is affected the most by selective capture. This increases the path length considerably and, consequently, reduces path reliability. Despite these two factors contributing to a shorter simulation MTTF value compared with the counterpart analytical MTTF value, we see that analytical model accurately predicts the best $(m_{p,} m_s)$ under which MTTF is maximized.

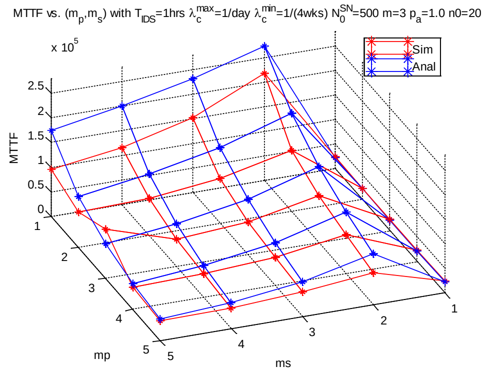


Fig. 7: Simulation vs. Analytical Results under Random Capture.

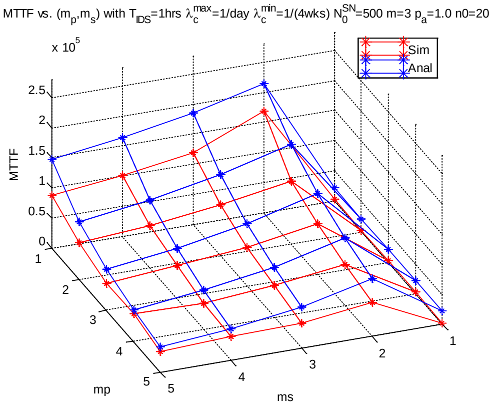


Fig. 8: Simulation vs. Analytical Results under Selective Capture.

8 Conclusion

In this paper we proposed and analyzed adaptive network defense management with three defenses for coping with selective capture and smart attack aiming to create holes near the base station in a wireless sensor network to block data delivery. Through numerical analysis, we demonstrated that our defenses are effective against selective capture and smart attack. There exist best protocol settings in terms of the best radio adjustment, the best redundancy level for multipath routing, the best number of voters, and the best intrusion invocation interval used for intrusion detection to maximize the system lifetime. Leveraging the analysis techniques developed in this paper, one can obtain optimal protocol settings at design time, store them in a table, and apply a simple table lookup operation at runtime with interpolation [23], [24], [25] to determine optimal settings for adaptive network defense management to maximize the system lifetime without runtime complexity.

This paper considers three defenses against selective capture attacks. For future work, we plan to consider selective deployment, i.e., populating more critical nodes than edge nodes to effectively counter selective capture. We also plan to further refine the inside attacker behavior model [2], [48] and extend the analysis to homogeneous or heterogeneous clustered WSNs [21], [26] without a BS. Finally, we also plan to investigate the use of trust/reputation management [13], [27], [47] to strengthen intrusion detection through "weighted voting," leveraging the knowledge of trust/reputation of neighbor nodes, as well as to tackle the "what paths to use" problem in multipath routing for intrusion tolerance in WSNs.

Acknowledgment

This work is supported in part by the U. S. Army Research Office under contract number W911NF-12-1-0445.

References

[1] R. Mitchell, and I.R. Chen, "A Survey of Intrusion Detection in Wireless Networks," Computer Communications, vol. 42, April 2014, pp. 1-23.

[2] R. Mitchell and I. R. Chen, "Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems," IEEE Transactions on Reliability, vol. 62, no. 1, pp. 199-210, 2013.

[3] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and defenses," 1st IEEE Int. Workshop on Sensor Network Protocols and Applications, 2003, pp. 113-127.

[4] I. M. Atakli, H. Hu, Y. Chen, W. S. Ku, and Z. Su, "Malicious node detection in wireless sensor networks using weighted trust evaluation," Spring simulation multiconference, 2008, pp. 836-843.

[5] G. Bravos and A. G. Kanatas, "Energy consumption and trade-offs on wireless sensor networks," 16th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications, 2005, pp. 1279-1283.

[6] S. Qun, "Power Management in Networked Sensor Radios A Network Energy Model," IEEE Sensors Applications Symp., 2007, pp. 1-5.

[7] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE Trans. Mobile Comput., vol. 3, no. 4, pp. 366-379, 2004.

[8] Y. Lan, L. Lei, and G. Fuxiang, "A multipath secure routing protocol based on malicious node detection," Chinese Control and Decision Conference, 2009, pp. 4323-4328.

[9] D. Somasundaram and R. Marimuthu, "A Multipath Reliable Routing for detection and isolation of malicious nodes in MANET," International Conf. on Computing, Communication and Networking, 2008, pp. 1-8.

[10] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," High Speed Networks, vol. 15, no. 1, pp. 33-51, 2006.

[11] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," IEEE Wireless Communications, vol. 15, no. 4, pp. 34-40, 2008.

[12] A.P.R. daSilva, et al., "Decentralized intrusion detection in wireless sensor networks," 1st ACM Workshop on Quality of Service & Security in Wireless and Mobile Networks, Montreal, Quebec, Canada, 2005.

[13] F. Bao, I. R. Chen, M. Chang, and J. H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection," IEEE Trans. Network and Service Management, vol. 9, no. 2, pp. 161-183, 2012.

[14] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communication, vol. 1, no. 4, pp. 660-670, 2002.

[15] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," 2nd ACM workshop on Security of ad hoc and sensor networks, Washington DC, USA, 2004.

[16] D. Huang and D. Medhi, "Secure pairwise key establishment in large-scale sensor networks: An area partitioning and multigroup key predistribution approach," ACM Transactions on Sensor Networks, vol. 3, no. 3, 2007, article no. 16.

[17] J. Ni, L. Zhou, and C. V. Ravishankar, "Dealing with random and selective attacks in wireless sensor systems," ACM Transactions on Sensor Networks, vol. 6, no. 2, 2010, article no. 15.

[18] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," IEEE Conference on Dependable Systems and Networks, pp. 637-646, 2004.

[19] W. Wang, M. Chatterjee, and K. Kwiat, "Coexistence with malicious nodes: A game theoretic approach," International Conference on Game Theory for Networks, pp. 277-286, 2009.

[20] R. Mitchell and I. R. Chen, "Behavior-Rule Based Intrusion Detection Systems for Safety Critical Smart Grid Applications," IEEE Transactions on Smart Grid, vol. 4, no. 3, Sept. 2013, pp. 1254 - 1263.

[21] H. Al-Hamadi and I. R. Chen, "Redundancy Management of Multipath Routing for Intrusion Tolerance in Heterogeneous Wireless Sensor Networks," IEEE Transactions on Network and Service Management, vol. 10, no. 2, pp. 189-203, 2013.

[22] I. R. Chen, A. P. Speer, and M. Eltoweissy, "Adaptive Fault-Tolerant QoS Control Algorithms for Maximizing System Lifetime of Query-Based Wireless Sensor Networks," IEEE Trans. on Dependable and Secure Computing, vol. 8, no. 2, pp. 161-176, 2011.

[23] T. Zhu, A. Mohaisen, Y. Ping, and D. Towsley, "DEOS: Dynamic energy-oriented scheduling for sustainable wireless sensor networks," IEEE INFOCOM, 2012, pp. 2363-2371.

[24] L. E. Bengtsson, "Lookup Table Optimization for Sensor Linearization in Small Embedded Systems," Journal of Sensor Technology, vol. 2, no. 4, pp. 177-184, 2012.

[25] J. Hedley, C. Roelfsema, and S. R. Phinn, "Efficient radiative transfer model inversion for remote sensing applications," Remote Sensing of Environment, vol. 113, no. 11, 2009, pp. 2527-2532.

[26] H. Al-Hamadi and I.R. Chen, "Integrated Intrusion Detection and Tolerance in Homogeneous Clustered Sensor Networks," ACM Transactions on Sensor Networks, vol. 11, no. 3, March 2015, article no. 47.

[27] I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 5, 2014, pp. 1200-1210.

[28] R. Mitchell and I.R. Chen, "Behavior Rule Specification-based Intrusion Detection for Safety Critical Medical Cyber Physical Systems," IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 1, 2015, pp. 16-30.

[29] H. Al-Hamadi, and I.R. Chen, "Countering Selective Capture in Wireless Sensor Networks," 10th IFIP/IEEE Conference on Network and Service Management, Zürich, Switzerland, 2013.

[30] S. Bandyopadhyay, and. E.J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," IEEE INFOCOM, 2003, pp. 1713-1723.

[31] P.J. Wan and C.W. Yi, "Coverage by Randomly Deployed Wireless Sensor Networks," IEEE/ACM Trans. Networking, vol. 14, no. 6, 2006, pp. 2658-2669.

[32] X. Li, D.K. Hunter, and S. Zuyev, "Coverage Properties of the Target Area in Wireless Sensor Networks," IEEE Trans. Information Theory, vol. 58, no. 1, 2012, pp. 430-437.

[33] D. Moltchanov, "Distance distributions in random networks," Ad Hoc Networks, vol. 10, no. 6, 2012, pp. 1146–1166.

[34] Y. Ebrahimi, and M. Younis, "Increasing Transmission Power for Higher Base-station Anonymity in Wireless Sensor Networks," IEEE ICC, Kyoto, Japan, June 2011.

[35] J.-P. Sheu, C. Jiang, and J. Deng, "An Anonymous Path Routing (APR) Protocol for Wireless Sensor Networks," IEEE International Conference on Wireless Communications, Networking and Information Security, June 2010.

[36] S. Seys and B. Preneel, "ARM: Anonymous Routing Protocol for Mobile Ad hoc Networks," 20th IEEE Conference on Advanced Information Networking and Applications, April 2006.

[37] A. Nezhad, A. Miri, and D. Makrakis, "Location privacy and anonymity preserving routing for wireless sensor networks", Int'l Journal of Computer and Telecom. Networking, vol. 52, no. 18, Dec. 2008.

[38] K. Mehta, D. Liu, and M. Wright, "Protecting location privacy in sensor networks against a global eavesdropper," IEEE Trans. on Mobile Computing, vol. 11, no. 2, 2012, pp. 320–336.

[39] D. Ying, D. Makrakis, and H.T. Mouftah, "Anti-traffic analysis attack for location privacy in WSNs," EURASIP Journal on Wireless Comm. and Networking, 2014, article 131.

[40] H. Park, et al., "PASSAGES: Preserving Anonymity of Sources and Sinks against Global Eavesdroppers," IEEE INFOCOM, April 2013.

[41] Q. Gu, X. Chen, Z. Jiang, and J. Wu, "Sink-Anonymity Mobility Control in Wireless Sensor Networks," IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Oct. 2009.

[42] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "A novel scheme for protecting receiver's location privacy in wireless sensor networks," IEEE Trans. on Wireless Comm., vol. 7, no. 10, 2008, pp. 3769-3779.

[43] H. Yu and M. Guo, "An efficient oil and gas pipeline monitoring systems based on wireless sensor networks," Inter. Conf. Information Security and Intelligence Control, 2012, pp. 178-181.

[44] M. R. Akhondi, A. Talevski, S. Carlsen, and S. Petersen, "Applications of wireless sensor networks in the oil, gas and resources industries," 24th IEEE International Conf. Advanced Information Networking and Applications, 2010, pp. 941-948.

[45] R. Lin, Z. Wang, and Y. Sun, "Wireless sensor networks solutions for real time monitoring of nuclear power plant," 5th World Congress on Intelligent Control and Automation, 2004, pp. 3663-3667.

[46] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "CitySee: urban CO2 monitoring with sensors," IEEE INFOCOM, 2012, pp. 1611-1619.

[47] F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Trust-based intrusion detection in wireless sensor networks," IEEE Inter. Conf. on Communications, Kyoto, Japan, June 2011, pp. 1-6.

[48] R. Mitchell, and I.R. Chen, "Modeling and Analysis of Attacks and Counter Defense Mechanisms for Cyber Physical Systems," IEEE Transactions on Reliability, 2015.

AUTHOR BIOGRAPHIES

Hamid Al-Hamadi received the Bachelor degree in Information Technology from Griffith University, Brisbane, Australia in 2003, the Master degree in Information Technology from Queensland University of Technology, Brisbane, Australia in 2005, and the PhD degree in Computer Science from Virginia Tech, USA, in 2014. His research interests include security, Internet of things, mobile cloud, wireless sensor networks, and reliability and performance analysis. Currently he is an assistance professor in the Department of Computer Science, Kuwait University, Khaldiya, Kuwait.

Ing-Ray Chen received the BS degree from the National Taiwan University, and the MS and PhD degrees in computer science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, wireless systems, security, trust management, and reliability and performance analysis. Dr. Chen currently serves as an editor for *IEEE Communications Letters, IEEE Transactions on Network and Service Management, The Computer Journal*, and *Security and Network Communications.*