Defending against Physical Attacks in Sensor Networks

Wenjun Gu, Xun Wang, Sriram Chellappan and Dong Xuan

***Abstract*— In this paper we study the defense of sensor networks against *Search-based Physical Attacks*. In search-based physical attacks, the attacker walks through the sensor network using signal detecting equipment to locate active sensors, and then destroys them. We design an effective defense approach to defend sensor networks against search-based physical attacks. The core principle of our defense is to trade short term local coverage for long term global coverage through the *Sacrificial Node*-assisted attack notification and states switching of sensors. The performance metric we use is *Accumulative Coverage*, which effectively captures coverage and lifetime of the sensor networks to measure sensor network performance. Our performance data clearly demonstrate that search-based physical attacks cause a significant deterioration in accumulative coverage. Our performance data also show that our defense approach can significantly decrease losses in accumulative coverage even under intense search-based physical attacks. To the best of our knowledge, ours is the first work that identifies the problem of search-based physical attacks and proposes defenses against it. We strongly believe that the viability of sensor networks in the future is contingent on their ability to resist physical attacks, which is the core of our work here.**

I. Introduction

Security in the sensor networks has been one of the research focuses in Wireless Sensor Networks (WSN) in these years. Research in this area has contributed a host of potential attacks in sensor networks and effective defenses against such attacks [1], [2], [3], [4], [5], [6], [7], [8]. It is widely accepted that viability of sensor network applications in the future is closely contingent on the security of the networks.

The small form factor of sensors, coupled with the unattended and distributed nature of their deployment expose sensor networks to a special class of attacks that could result in the physical destruction of sensors. We denote *Physical Attacks* as those that result in the physical destruction of sensors, thereby rendering them permanently nonoperational. The significance of studying physical attacks comes from the following factors. Physical attacks are *inevitable* threats in sensor networks. Physical attacks are relatively simple to launch and *fatal* in destruction. In the simplest case, the attacker can just drive a vehicle in the sensor field or hurl grenades/bombs in the field and destroy the sensors. A smarter attacker can detect and destroy sensors with stealth by moving across the sensor network. In any case, the end result of physical attacks can be quite destructive. The backbone of the network (the sensors themselves) is destroyed. Destruction of sensors may also result in the violation of the network paradigms. This could be the topology, routing structure, power dissipated etc. As such, a wide spectrum of impacts may result due to physical attacks and when left unaddressed, physical attacks have the potential to render the entire sensor network mission useless.

Wenjun Gu, Xun Wang, Sriram Chellappan and Dong Xuan are with The Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210, U.S.A. E-mail: {gu, wangxu, chellapp, xuan, lai}@cse.ohio-state.edu.

Our focus in this paper is *Search-based Physical Attacks*. We define search-based physical attacks as those that *intelligently* search for sensors. The searching process is executed by means of detecting signals emitted by the sensors. Once sensors are identified, the attacker physically destroys the sensors. This process is opposed to a rather blind or brute force destruction of sensors in the field (using bombs, grenades, tanks etc) that will cause casualties to the deployment field, which the attacker might want to preserve (airports, oil fields, battlefields etc. of interest to the attacker).

In this paper, we first introduce a representative search-based physical attacks model. In our attack model, the attacker continuously locates sensors by means of signal detection and physically destroys the detected sensors. We then propose a *Sacrificial Node*-assisted defense protocol to defend sensor networks against search-based physical attacks. The core principle of our defense is to trade short term local coverage for long term global coverage through the *sacrificial node*-assisted attack notification and states switching of sensors. A *sacrificial node* is the one which detects the attacker for other sensors at the risk of its being detected and destroyed by the attacker. The existence of *sacrificial nodes* compensates the weakness of the sensors' ability to detect the attacker. Our defense protocol does not assume a priori knowledge about the attacker. Our performance metric in this paper is the *Accumulative Coverage* of the network. Accumulative Coverage captures both the lifetime and coverage and as such is an effective metric to measure performance. Our performance data clearly show that search-based physical attacks dramatically reduce accumulative coverage. However, with our defense mechanism in place, the accumulative coverage can be improved significantly even under intense attacks.

Physical attacks are patent and potent threats to sensor networks. We believe that the viability of future sensor networks is contingent on their ability to resist physical attacks. As such, our work is an important first step in this regard. The defense strategy we propose is novel, simple and effectively defends against search-based physical attacks. The rest of the paper is organized as follows. In Section II, we discuss the attack model. Section III describes our defense protocol and Section IV reports the performance evaluation. We present related work in Section V, and conclude our work in Section VI.

II. Modeling Search-based Physical Attacks

In this section, we first provide a general description of physical attacks, followed by detailed discussions on search-based physical attacks, which is the main focus of this paper.

A. Physical Attacks in Sensor Networks

Physical attacks are those attacks that result in the physical destruction of the sensors, rendering them permanently non-operational. A wide spectrum of physical attacks is possible in the domain of sensor networks. Broadly speaking, the entire spectrum of physical attacks can be considered to operate in two phases, namely the *targeting* phase and the *destruction* phase. In the targeting phase, the attacker tries to identify the sensors or the deployment area of the sensor network. Then, the destruction phase follows to destroy the sensors. Based on the above discussions, we classify physical attacks into two classes.

Blind Physical Attacks: In blind physical attacks, the execution of the targeting phase is to just identify the sensor deployment field. Following this, the identified sensor deployment field is under physical attack using a brute-force approach. Typical brute-force physical destruction instances include physical attacks in the form of bombs/grenades, tanks/vehicles driven around destroying contiguous portions of sensor deployment field. Sensors that happen to be in the vicinity of destruction areas are destroyed.

Search-based Physical Attacks: Here the attacker first searches for sensors in the network by detecting signals emitted by the sensors using appropriate signal detecting equipment. After the detection, the attacker destroys the identified sensors physically. Destruction of the small size sensors is typically accomplished through physical force, heat, radiation and other hardware/circuit tampering techniques that in effect destroy the physical hardware.

Attackers indulge in search-based physical attacks by walking in the sensor network deployment area to search for and then physically destroy sensors. The execution of search-based physical attacks can be modeled as the execution of three actions namely *searching action*, *moving action*, and *destruction action*.

- *Searching action*: In the searching action, the attacker searches for sensors. There are three features characterizing this action, namely *target of search*, *method of search* and *ability of search*. The target of attacker's can be any sensor or some specific sensors such as cluster heads, data aggregators etc (depending on the network structure). The method of search can be detecting emitted signals, distinguishing radio frequencies, traffic pattern analysis etc. In general, signal detection is the most basic method to search for sensors. This depends on the type and strength of the signals emitted by the sensors. The attacker can also distinguish between radio frequencies to find sensors with different roles in the sensor network. It can also perform traffic analysis on sensors to find out important sensors like cluster heads, data aggregators etc. The searching ability can be the signal detection distance, detection accuracy.
- *Moving action*: The moving action can be following pre-programmed paths or it can be random, in the absence of a target. When there is only one target, the attacker moves towards it. When there are multiple targets detected, the attacker needs to schedule its movement to reach targets efficiently. The modeling features related to attacker's moving action include *moving intelligence* and *moving strength*. The moving intelligence includes attacker's motion sequence, such as the scheduling of movement, choice of targets etc. Moving strength includes the moving speed of attacker, total movement length, and the attacker's ability to overcome obstacles in the sensor fields.
- *Destruction action*: The attacker can use different methods to destroy sensors. The modeling features in destruction are *destruction method* and *destruction strength*. Destruction method includes the way the attacker physically destroys sensors. This includes physical force, heat, radiation and other hardware/circuit tampering techniques. Destruction method also depends on the accuracy of target location detection. If the target is accurately located, the destruction method is to destroy only the target in the location identified. However, if the detection is inaccurate, then the target is isolated within an area (instead of being located accurately). Then the destruction method is to destroy all the sensors in that area. Destruction strength measures how much time and/ or energy the attacker spends to destroy a sensor. In some cases, the destruction action is overlapped with the moving and searching action, i.e. the attacker has to move during destruction while searching for new targets as it moves.

These three actions can be overlapped in time in order to achieve efficient attack. For example, whenever the attacker moves, it keeps searching for sensors. The attacker can even keep searching and moving during destroying sensors.

In [9], we studied the issue of sensor network performance under blind physical attacks. There we studied the issue of deployment of sensors in a network to meet lifetime requirement under blind attacks. In this paper we focus on search-based physical attacks. In many situations, blinds attacks may be infeasible for the attacker, and the attacker will indulge in search-based physical attacks. For instance, in some cases it may be necessary for the attacker to preserve the field of interest. Such fields can include airports, oil fields, battlefields of the attacker side etc. Destroying the entire area by means of grenades or bombs may not be possible for the attacker as it will destroy the deployment field. In such cases search-based attacks will be used to destroy sensors. Also, the search-based attacks are more efficient in that they operate by stealth compared with blind physical attacks that use a brute force destruction approach. Our focus in this paper is search-based physical attacks. In the following, in Section II-B, we present our search-based physical attack model to be used in the rest of the paper.

B. Modeling Search-based Physical Attacks

In search-based physical attacks, the basic method the attacker uses to identify sensors is to detect signals emitted

by the sensors. We classify signals emitted by sensors into two types. Passive signals include heat, vibration, magnetic signals that are part of the physical characteristics of the sensors[1]. Active signals on the other hand include communication messages, beacons, query messages that are part of normal network communication paradigms. These two are quite different from the perspective of attacker detection. Passive signals are very small in range, and their detection can enable the attacker to accurately detect their source (the sensor emitting the passive signal). Active signals can propagate longer distance, but the attacker can only isolate the source of an active signal within an area. In case the attacker detects multiple sensors, if the attacker is equipped with memory, the attacker can store the locations of multiple sensors that it detects. Thus, if we use R_{ps} and R_{as} to denote the maximal distances from where the attacker can detect passive and active signals respectively, R_{ps} is smaller than R_{as}.

The ability of the attacker to detect a sensor also depends on the state of the sensor. A sensor is *Dead* if it has been physically destroyed by an attacker. Otherwise it is *Alive*. In our model, a sensor that is *Alive* can be in one of the following three states, namely *sleeping, sensing* and *sending* state. A sensor can voluntarily turn itself off and be in the *sleeping state*. In this state, the sensor emits no signal[2] and hence cannot be detected by the attacker. A sensor in the *sensing state* carries out only sensing tasks, without sending out any active signal. The signals emitted during sensing are just passive signals. A sensor in the *sending state* emits both passive and active signals. We call a sensor *Active* if it is in sensing state or sending state. An active sensor can be detected by the attacker by detecting the signals emitted by the sensor. A sensor can instantaneously switch among these three states at will as long as it is alive. In our model, the sensors are also capable of sensing the attackers[3]. However, in our model the attacker is more powerful than sensors. Thus the range within which an attacker detects a sensor is larger than that within which a sensor can detect the attacker.

Model 1 describes our search-based physical attack model. This model describes the attacker's response to different events taking place during the attack process. Initially, the attacker does not have any target sensor to destroy. Here, the attacker performs a random straight line walk in the network field and keeps detecting passive or active signals. We use v to denote the moving speed of the attacker [4].

Once the attacker detects a signal from a sensor, the attacker first checks the type of signal used to detect the sensor; Case 1: If the signal was a passive signal, the attacker first estimated the location of the source of the signal. If the attacker has no target, it then sets the sensor that emitted this signal as the target and walks towards it. Otherwise, if the attacker already has a target which was detected through a passive signal, it immediately puts the source of this signal into memory. If the attacker has a current target detected through an active signal, the attacker puts the current target into memory and sets the newly detected sensor (through a passive signal) to be the target. Case 2: If the signal detected was an active signal, the attacker identifies a *sweeping area* and puts the area in memory. We refer this area as *sweeping area*. Precisely, the sweeping area is the small area within which the attacker isolates sensors detected by means of an active signal. Obviously, closer the detected sensor (stronger active signal detected) is, smaller is the sweeping area. If the attacker has no target when this active signal is detected, the attacker sets the sensor that emitted this signal to be the current target and walks towards it. If the attacker already has a target, it will put this newly detected sensor and the corresponding sweeping area into memory. In our model, the attacker at any point in time can have only one sensor as a target to destroy. Multiple detected targets/ sweeping areas can be put into memory for future targets.

During the movement, if there is no new signal detected, the attacker either keeps walking randomly if it has no target or keeps moving to the target. Once the attacker reaches the target, the attacker will destroy the target. There are two cases in which the attacker enters this situation. Case 1: If the detection of the current attack target is through passive signal, the attacker carries out physical destruction directly at the target location which is identified when the target was detected. Case 2: If the detection of the current attack target was through an active signal, the attacker *sweeps* once it reaches the sweeping area. Specifically, it sweeps across this sweeping area destroying any sensor in that area. In some

Model 1 Search-based physical attack model

1: Initialization: $Target \leftarrow \Phi$; $Mem \leftarrow \Phi$;
2: **while** the attacker is alive **do**
3: **switch** type of event
4: **case** detecting a sensor S through passive signal:
5: $Target = \Phi$: $Target \leftarrow S$; $Target.type \leftarrow passive$; $Target.location \leftarrow Location\ of\ S$;
6: $Target \neq \Phi$ **AND** $Target.type = passive$: add S to Mem;
7: $Target \neq \Phi$ **AND** $Target.type = active$: add $Target$ to Mem; $Target \leftarrow S$; $Target.type \leftarrow passive$; $Target.location \leftarrow Location\ of\ S$;
8: **case** detecting a sensor S through active signal:
9: $Target = \Phi$: $Target \leftarrow S$; $Target.type \leftarrow active$; $Target.location \leftarrow Sweeping\ area\ of\ S$;
10: $Target \neq \Phi$: add S to Mem;
11: **case** reaching $Target.location$:
12: $Target.type = passive$: Directly destroy $Target$; $Target \leftarrow Remove(Mem)$;
13: $Target.type = active$: Sweep the sweeping area of $Target$; $Target \leftarrow Remove(Mem)$;
14: **default:**
15: Whenever $Target \neq \Phi$, walk towards $Target.location$, otherwise walk randomly;
16: **endswitch**
17: **end while**

[1]We assume the sensors are camouflaged from the attacker and the attacker will not be able to visually identify sensors. Thus the attacker cannot use visual signals to detect sensors.

[2]We assume that in the sleeping state, even if minute signals are emitted, they are imperceptible to the attacker.

[3]In the worst case, a sensor can estimate the attackers location just prior to being physically destroyed by the attacker.

[4]If the attacker reaches the boundary of the network, it is aware of the fact and turns in a suitable direction in order to once again walk into the network.

model instances, the attacker can keep detecting new sensors during the sweeping, and can then put the newly detected sensors in it memory. We also consider this case during our performance evaluation.

Once the attacker has physically destroyed a sensor, if the attacker is equipped with memory, it checks its memory. If the attacker has some sensors in memory, it will pick the closest sensor detected by a passive signal as a target as long as there is one. Otherwise, it chooses the nearest sensor detected by active signals as the next target. If the memory content is empty, the attacker does a random straight line walk to search for sensors.

C. Discussions

We wish to state that our search-based attack model presented here is a representative instance of the general model. It can be extended to represent a wide spectrum of search-based physical attacks. One extreme case is; if there is no search stage in the attack, the attacker can just use random sweeping in order to destroy sensors, which is similar to brute force attacks. The other extreme cases are attacks that destroy only specific sensors among many targets. Such sensors can be cluster heads, data aggregators etc. in the sensor network. Obviously, in the latter case we are dealing with a very sophisticated attacker. Another example of sophisticated physical attacks is that the attacker aims to destroy the functionality of the sensor network by partitioning the sensing field. This attack may be hard to achieve in the sense that the attacker needs a priori knowledge of the topology and architecture of the sensor field. Besides, this attack may not be efficient when there exist some sensors with much larger communication range such as data aggregators. In our current model, we assume there is only one attacker. Our current attack model can be easily extended to multiple attackers if there is no cooperation among the attackers. Otherwise, the attack model and the defense algorithm should take into account the efficient cooperation among attackers. The study of sophisticated physical attacks and the cooperation among multiple attackers are part of our on-going work.

A specific type of attack most related to physical attacks is jamming attacks [8], where the attacker jams or interferes with the radio frequencies that sensor(s) are using. Physical attacks are quite different from jamming attacks in that jamming only causes a loss of operation for the attack duration, while physical attacks result in irreversible sensor destructions. Furthermore, the standard defense for jamming attacks using forms of spread-spectrum communication cannot be used here as the attacker just needs raw signals to detect sensors.

III. Defending against Search-based Physical Attack

With the model on search-based physical attacks in place, we will discuss defending sensor network against search-based physical attacks. We will first give the design rationale followed by detailed description of our defense protocol.

A. Design Rationale

The primary success criteria of the attacker in conducting search-based physical attacks are the number of sensors it can destroy and the coverage loss incurred by those destroyed sensors. As such, it becomes clear that any defense strategy against search-based physical attacks should not only focus on protecting individual sensors from being destroyed, but also distributing the destroyed sensors as uniformly as possible, which helps to prevent large coverage loss or even network disconnectivity. While a simple defense strategy is to let all sensors sleep to avoid being detected, one has to keep in mind the performance of the sensor networks. An ideal defense algorithm should minimize sensor detection by the attacker with little compromise on the overall performance of the sensor network.

In order to evaluate the performance of sensor networks under search-based physical attacks, we define a novel metric in this paper, namely *Accumulative Coverage (AC)*. AC is defined as the integration of the network coverage over the *effective lifetime* of the sensor network. Network coverage is defined as the percentage of the sensor field that is in the sensing range of at least one active sensor [5], and effective lifetime is the time period until when the sensor network becomes nonfunctional because the coverage falls below a certain threshold α. Denoting *coverage(t)* as the network coverage at time t, and EL, as the effective lifetime, we have,

$$AC = \int_{t=0}^{EL} coverage(t)dt. \quad (1)$$

We believe that AC is an effective metric to measure the performance of a sensor network in many situations since it effectively combines both coverage and lifetime, two of the most important performance metrics in sensor networks. A general metric commonly used in the literature is effective lifetime. Effective lifetime is defined as the maximum time period during which the coverage is above a certain threshold and thus considers both coverage and lifetime. However, it is not representative enough for situations where for the same effective lifetime, a sensor network with a high coverage can provide more accurate information than one with a lower coverage. Our metric, AC not only considers coverage threshold and lifetime, but is also more representative of real life situations. Thus AC is the basic metric we use to evaluate the performance of a sensor network under search-based physical attacks.

Our objective is to maximize AC under search-based physical attacks. Our rationale is to prevent sensors' active and passive signals from being detected by the attacker. Our objective is to design a distributed defense protocol using only local information. To do so, we first need to let sensors in the local vicinity of the attacker switch from sending state to sensing/sleeping state in time in order to escape detection by the attacker. This can minimize the sensors' detection and destruction by the attacker and increase their lifetime. On

[5]We assume 1-coverage in this paper. In some situations, each point in the sensor field needs to be covered by more than 1 sensor (say k). This is called k-coverage.

TABLE I
NOTATIONS AND DEFINITIONS

Notation	Definition
AC	Accumulative Coverage
EL	Effective Lifetime
$Coverage(t)$	Network coverage at time t
α	Network coverage threshold
N	Number of sensors in the network
S	Area of the sensor field
f	Active signal frequency
R_{noti}	Notification range
R_{as}	Active signal detection range
R_{ps}	Passive signal detection range
R_a	Sensor's detection range for attacker
R_s	Sensor's sensing range for coverage
v	Attacker speed
M	Attacker memory size
k_i	Number of sensors in sensor s_i's protection area
k_i'	Number of unprotected sensors among k_i
$d(i,j)$	Distance between sensor s_i and sensor s_j
$u(i)$	Utility value of sensor s_i
$u_j(i)$	Contribution of sensor s_j to $u(i)$
U_{th}	Utility threshold
U_{ref}	Reference utility value
$D(i)$	Timer for SN message of sensor s_i
$T_1(i)$	Timer from sleeping to sensing for sensor s_i
$T_2(i)$	Timer from sleeping to sending for sensor s_i
$T_3(i)$	Timer from sensing to sending for sensor s_i

the other hand, we need to keep the sensors that are not in the neighborhood of the attacker in sensing/sending state to maintain necessary coverage. In our defense protocol, a sensor detects the attacker by the signals it emits (vibration, motion etc) and/or its physical properties (electromagnetic, metal etc). The sensor will send out an attack notification message notifying all the active sensors in its local notification area before it is destroyed by the attacker. The notification message contains the position of the attacker. Recipient sensors may switch to sensing/sleeping state in a timely manner when the attacker approaches in order to avoid being detected by the attacker. They will then switch back to sensing/sending state independently in a conservative manner when the attacker leaves.

While it is obvious that the sensors can protect them from being detected by switching to sensing/sleeping states after receiving the attack notification message, we observe that this may not be always optimal in a global point of view. The sensors outside the local notification area of the destroyed sensor will not be aware of the approaching of the attacker, which makes them in danger. In this situation, it will be better for a few sensor that receive the attack notification message to keep in active state so that they can inform other unaware sensors before they are detected. We call these sensor nodes *sacrificial nodes*. They could have protected themselves by switching to sleeping state, however their sacrifice helps to protect more other sensors, especially when the local area is relatively dense. The challenge is how those sensors can decide whether they should be *sacrificial nodes*, which will be described in detail in the following sebsections.

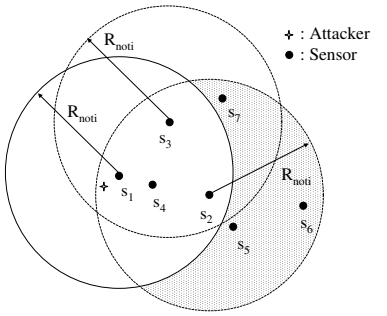


Fig. 1. Protocol description.

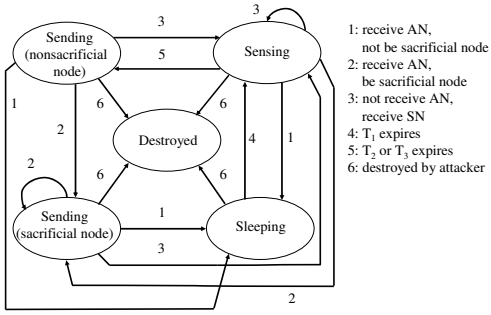


Fig. 2. States switching and events.

B. Defense Protocol

In this section, we first discuss our localized defense protocol in detail. We then discribe the mechanism used to decide *sacrificial nodes*. All the notations used in this paper are listed in Table I.

1) Protocol Description: The protocol is executed by individual sensors switching among different states triggered by one of several events, which is shown in Fig. 2. The arrows in Fig. 2 denote states switching among different states, while the digital numbers beside the arrows denote the events that trigger the corresponding states switching. At the beginning, one active sensor detects the attacker. It sends out an attack notification message (AN message) and stays in sending state. Those active sensors receiving the AN message will decide whether to be *sacrificial nodes* or not based on *sacrificial nodes* selection mechanism. For the recipient sensors of the AN message that decide to be *nonsacrificial nodes*, they will calculate two timers, T_1 and T_2, and switch to sleeping state immediately, which corresponds to event 1 in Fig. 2. These sensors will switch back to sensing and sending states as *nonsacrificial nodes* after T_1 and T_2 expire respectively, which corresponds to event 4 and 5 in Fig. 2. For other recipient sensors of the AN message who decide to be *sacrificial nodes*, they will send out *sacrificial node* notification messages (SN messages) and stay in sending state, which corresponds to event 2 in Fig. 2. For the sensors that do not receive the original AN message but receive at least one corresponding SN messages, they will calculate a timer T_3 and switch to sensing state immediately, which corresponds to event 3 in Fig. 2. These sensors will switch back to sending state as *nonsacrificial nodes* after T_3 expires, which corresponds to

event 5 in Fig. 2. Obviously, the sensors that are destroyed by the attacker will switch to destroyed state, which corresponds to event 6 in Fig. 2.

The AN message contains the global ID of the sensor that sends out this message, while the SN message contains the global ID of both the sensor sending out this SN message and the sensor sending out the corresponding AN message. The detailed description of *sacrificial nodes* selection scheme is discussed in Section III-C. The discussion of the timers, T_1, T_2 and T_3, will be detailed in Section III-D.

2) Example: In the following, we use an example in Fig. 1 to further explain our defense protocol described above. Before detecting or being notified of the approaching of the attacker, all sensors are in sending state as *nonsacrificial nodes*. Suppose at some time, sensor s_1 in Fig. 1 detects the attacker and sends an AN message to all other sensors in its notification area. The notification message contains the global ID of sensor s_1 and the notification area is a circle of radius R_{noti} centered at sensor s_1. In our defense protocol, we let R_{noti} be the same as the communication range for the sensors. Recall that the attacker is generally more powerful than a sensor in terms of sensing ability, as such, sensor s_1 is quite likely to be detected by the attacker already when it detects the attacker. Thus it is better for sensor s_1 to send out AN message instead of switching to sensing/sleeping state. After sending out the AN message, sensor s_1 will stay in sending state.

For the recipients of the AN message sent by sensor s_1, which are sensors s_2, s_3 and s_4 in Fig. 1, we assume sensors s_2 and s_3 decide to be *sacrificial nodes* while sensor s_4 does not. Sensors s_2 and s_3 will each send out an SN message at different time. In our protocol, we apply a randomized algorithm to let different *sacrificial nodes* send out SN messages at different time, thus alleviating the problem of message collision, the detail of which is discussed in Section III-C. After sensors s_2 and s_3 send out the SN messages, they will stay in sending state as *sacrificial nodes*. The SN message of sensor s_2 contains the global ID of sensor s_2 and sensor s_1. The usage of this message is two folded. First, it is used to update its state information stored in its neighbors, the usage of which will be described in Section III-C. Second, it is used by the sensors in its protection area for states switching, which will be described below. On the other hand, sensor s_4 will calculate two timers, T_1 and T_2, and switch to sleeping state immediately. After T_1 and T_2 expire, sensor s_4 will switch back to sensing and sending (as *nonsacrificial node*) states respectively.

In Fig. 1, sensors s_5, s_6 and s_7 receive the SN message sent by sensor s_2, but they did not receive the corresponding AN message sent by sensor s_1. They each will independently calculate a timer T_3 and switch to sensing state immediately. By doing so, they are protected from being detected via active signals since the attacker may approach them in the near future. However, it may not be good for these sensors to switch to sleeping state for two reasons. First, this will result in a large coverage loss, which is an overkill since the attacker will only choose to move in one direction after destroying sensor s_1. Second, they are already in the protection area of sensor s_2. They may be notified of the approaching of the attacker by sensor s_2 before their own passive signals are detected and then switch to sleeping state in time.

C. Sacrificial Nodes Selection

1) Derivation of Utility Function $u(i)$: Intuitively, a sensor is more preferable to be a *sacrificial node* when there exist more sensors in its protection area. In Fig. 1, sensor s_2 is more preferable than sensor s_4 because it can potentially protect more other sensors. The protection area of sensor s_4 and other sensors in the shaded area are not shown in Fig. 1 for clarity. Thus, a simple utility function that can be used to measure the preference of a sensor s_i being a *sacrificial node* is given by,

$$u(i) = k_i, \tag{2}$$

in which k_i denotes the number of sensors in the protection area of sensor s_i.

It may seem obvious that a sensor with high utility value is always preferable to be a *sacrificial node*. However, if two sensors both have high utility values and they are close to each other, it is not preferable for both of them to be *sacrificial nodes*. The reason is that the protection areas of both sensors have much overlap due to the fact that they are close to each other. Selecting the second one as a *sacrificial node* brings little extra benefit. Besides, it incurs more risk to select both as *sacrificial nodes* since both of them become potential targets for the attacker now instead of one. A reasonable modification is given by,

$$u(i) = k_i^{'}, \tag{3}$$

in which $k_i^{'}$ denotes the number of sensors in the protection area of sensor s_i that are not in the protection area of any other *sacrificial node*. If we assume sensor s_3 in Fig. 1 is a *sacrificial node*, which is known by sensor s_2 via the SN message of sensor s_3, sensor s_7 will not be counted in the calculation of the utility function for sensor s_2.

Furthermore, we observe that the relative distance of the sensors in the protection area also makes difference. In Fig. 1, let's assume the attacker moves to the right after it destroys sensor s_1. Compared with sensor s_6, sensor s_5, which is closer to sensor s_2, is more likely to be detected before sensor s_2 detects the attacker and sends out an AN message. In this case, the contribution of sensor s_5 to the utility function of s_2 is smaller than that of sensor s_6. Recall that the sensors have no knowledge of the sensing range of the attacker, therefore we weighted the contribution of sensor s_j to $u(i)$, denoted as $u_j(i)$, by the distance between sensor s_i and sensor s_j, denoted by $d(i,j)$. Thus we obtain the following utility function,

$$u(i) = \sum_{s_j \in k_i^{'}} u_j(i) = \sum_{s_j \in k_i^{'}} \frac{d(i,j)}{R_{noti}}. \tag{4}$$

Here, we also use $k_i^{'}$ to denote the set of sensors that are in the protection area of sensor s_i but not in the protection area of any other *sacrificial node* for clarity.

In the ideal situation, assuming all sensors have full knowledge about the attacker including R_{as} and R_{ps}, a sensor s_i is able to calculate which of its $k_i^{'}$ neighbors are already

detected by the attacker and should not be considered in $u(i)$. We denote the ideal $u(i)$ assuming full knowledge of the attacker as $u^{opt}(i)$ and denote $u_j^{opt}(i)$ as the optimal $u_j(i)$. In the following, we will prove that the utility function in (4) is optimal in terms of minimizing the expected mean square error between $u(i)$ and $u^{opt}(i)$ under the assumption that the sensors have no a priori knowledge of the sensing ability of the attacker.

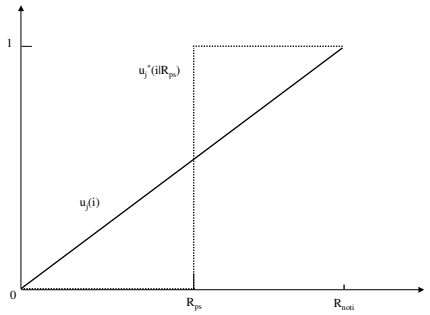


Fig. 3. Proof of Lemma 1.

Lemma 1: $u_j(i) = \frac{d(i,j)}{R_{noti}}$ is optimal among all functions of the form $u_j(i) = F(d(i,j))$ in terms of minimizing the expected mean square error between $F(d(i,j))$ and $u_j^{opt}(i)$.

Proof: In the ideal case assuming the sensors have full knowledge of the sensing ability of the attackers, specifically, the values of R_{as} and R_{ps}, the optimal contribution of sensor s_j to the utility function of sensor i, denoted by $u_j^{opt}(i) = u_j(i|R_{as}, R_{ps})$ is shown in Fig. 3. The sensors in set $k_i^{'}$ with distance $d(i,j)$ smaller than R_{ps} have already been detected before sensor s_i detects the attacker and sends out the attack notification message, so these sensors do not need to be protected and considered in the utility function. All other sensors in set $k_i^{'}$ are at risk and thus need to be considered. However, the sensors have no a priori knowledge of R_{ps}. It is reasonable to assume the value of R_{ps} follows a uniform distribution between 0 and R_{noti}, which is a commonly used distribution for random variables we have no a priori knowledge about. The optimal $F(d(i,j))$ in this case should minimize the expected Mean Square Error (MSE), which is,

$$MSE(F(d(i,j))) = E[(F(d(i,j)) - u_j^{opt}(i))^2]$$

$$= \frac{1}{R_{noti}} \int_{R_{ps}=0}^{R_{noti}} [F(d(i,j)) - u_j(i|R_{as}, R_{ps})]^2 dR_{ps}$$

$$= \frac{1}{R_{noti}} (\int_{R_{ps}=0}^{d(i,j)} [F(d(i,j)) - 1]^2 dR_{ps} + \int_{R_{ps}=d(i,j)}^{R_{noti}} [F(d(i,j)) - 0]^2 dR_{ps})$$

$$= F(d(i,j))^2 - 2\frac{d(i,j)}{R_{noti}} F(d(i,j)) + \frac{d(i,j)}{R_{noti}}.$$

The above expected mean square error is minimized when $F(d(i,j))$ equals $d(i,j)/R_{noti}$, thus the lemma holds. ■

Applying Lemma 1, we can obtain the following theorem.

Theorem 1: The utility function $u(i) = \sum_{s_j \in k_i^{'}} \frac{d(i,j)}{R_{noti}}$ is optimal in terms of minimizing the expected mean square error between $u(i)$ and $u^{opt}(i)$.

Proof: We assume the utility function of node s_i is the sum of $u_j(i)$ for all $j \in k_i^{'}$. Therefore, the utility value for s_i is $u(i) = \sum_{s_j \in k_i^{'}} u_j(i)$ and the ideal utility value is $u^{opt}(i) = u(i|R_{as}, R_{ps}) = \sum_{s_j \in k_i^{'}} u_j(i|R_{as}, R_{ps})$. Denote the sensors in the set $k_i^{'}$ by $s_{j_1}, s_{j_2}, \ldots, s_{j_{|k_i^{'}|}}$. Without loss of generality, we assume $0 \le d(i,j_1) \le d(i,j_2) \le \ldots \le d(i,j_{|k_i^{'}|}) \le R_{noti}$. The expected Mean Square Error (MSE) between $u(i)$ and $u^{opt}(i)$ is,

$$MSE(u(i)) = E[(u(i) - u^{opt}(i))^2]$$

$$= \frac{1}{R_{noti}} \int_{R_{ps}=0}^{R_{noti}} [u(i) - u(i|R_{as}, R_{ps})]^2 dR_{ps}$$

$$= \frac{1}{R_{noti}} \int_{R_{ps}=0}^{R_{noti}} [\sum_{s_j \in k_i^{'}} u_j(i) - \sum_{s_j \in k_i^{'}} u_j(i|R_{as}, R_{ps})]^2 dR_{ps}$$

$$= \frac{1}{R_{noti}} (\int_{R_{ps}=0}^{d(i,j_1)} [\sum_{s_j \in k_i^{'}} u_j(i) - (|k_i^{'}| - 0)]^2 dR_{ps}$$

$$+ \int_{R_{ps}=d(i,j_1)}^{d(i,j_2)} [\sum_{s_j \in k_i^{'}} u_j(i) - (|k_i^{'}| - 1)]^2 dR_{ps} + \ldots$$

$$+ \int_{R_{ps}=d(i,j_{|k_i^{'}|})}^{R_{noti}} [\sum_{s_j \in k_i^{'}} u_j(i) - (|k_i^{'}| - |k_i^{'}|)]^2 dR_{ps})$$

$$= (\sum_{s_j \in k_i^{'}} u_j(i))^2 - 2\frac{\alpha}{R_{noti}} \sum_{s_j \in k_i^{'}} u_j(i) + \frac{\beta}{R_{noti}}.$$

In the above, α is given by $\sum_{s_j \in k_i^{'}} d(i,j)$ and β is given by $d(i,j_1)(2|k_i^{'}| - 1) + d(i,j_2)(2|k_i^{'}| - 3) + \ldots + d(i,j_{|k_i^{'}|})1$. By deriving the first and second derivatives, the above expected mean square error is minimized when $u(i) = \sum_{s_j \in k_i^{'}} u_j(i) = \sum_{s_j \in k_i^{'}} \frac{d(i,j)}{R_{noti}}$. Thus, the theorem holds. ■

In (4), if we replace $k_i^{'}$ by the average number of neighbors for a sensor and replace the weight by the maximum weight 1, we obtain approximate upper bound for $u(i)$, which is denoted by U_{ref}. The expression of U_{ref} is given by,

$$U_{ref} = \frac{N\pi R_{noti}^2}{S}, \tag{5}$$

in which N is the number of sensors in the network and S is the area of network. Since the value of $k_i^{'}$ is usually smaller than the average number of neighbors for a sensor and the weight is no more than 1, the utility value of a sensor is generally smaller than U_{ref}.

2) Sacrificial Nodes Selection Scheme: Individual sensors can calculate their utility values as described before. We now describe the criterion used by a sensor to decide whether it should be a *sacrificial node* or not based on its utility value. Intuitively, a sensor that has certain high utility value should become a *sacrificial node*, thus an empirical threshold U_{th} is necessary here. The sensors whose utility values are above U_{th} will become *sacrificial nodes*. The value of U_{th} lies in

the interval [0, U_{ref}]. Similar to the utility function, an ideal utility threshold is impossible to obtain without the knowledge of the attacker information. We will investigate the sensitivity of our defense protocol's effectiveness to U_{th} in Section IV and give guidelines in choosing a reasonable U_{th}.

After the discussion of *sacrificial node* selection criterion, we will describe the scheme used by the recipient sensors of the AN message for *sacrificial nodes* selection. Since it is possible that multiple sensors have initial utility values larger than U_{th}, we introduce a randomized algorithm here to prevent the collision of SN messages and deal with the problem of protection area overlap. After first calculating the utility function, the sensors whose utility values are smaller than U_{th} will switch to sleeping state. While other sensors, whose utility values are larger than or equal to U_{th}, also called candidate *sacrificial nodes*, will calculate a random delay and set a timer, denoted by $D(i)$, based on their utility values by,

$$D(i) = \begin{cases} \epsilon * \Delta t & , \quad u(i) \geq U_{ref} \\ \Delta t + (1 - \frac{u(i)}{U_{ref}}) * \Delta t, & U_{th} \leq u(i) < U_{ref} \end{cases} \quad (6)$$

in which ϵ is a random number uniformly distributed in [0,1] and Δt is an adjustable parameter. Ideally, Δt should be as small as possible to avoid a large delay of SN messages. However it should be comparable to the transmission time of an SN message to avoid collision among different SN messages. A candidate *sacrificial node* will send out an SN message after its timer expires and then become a *sacrificial node* formally. Thus, the sensor with higher utility value generally will send out SN message earlier. After receiving an SN message, a candidate *sacrificial node* who has not sent out its SN message will cancel its timer and adjust its utility value accordingly by (4). If the new value is less than U_{th}, it will switch to sleeping state. Otherwise, it will calculate a new delay and set a timer as above. Therefore, it is quite unlikely that multiple SN messages will collide with each other. This process iterates until each recipient sensor of the AN message either becomes a *sacrificial node* or switches to sleeping state.

D. States Switching Timers

Recall that the attacker will proceed to destroy other detected sensors in its memory or choose a random direction if its memory is empty, the sensors that receive the AN/SN messages cannot accurately predict the movement of the attacker. In the protocol described above, we let the sensors triggered by events 1 and 3 in Fig. 2 immediately switch to sleeping and sensing states respectively. We admit this is a conservative scheme. The sensors may switch to sensing/sleeping state too early or even unnecessarily if the attacker never approaches them, but this guarantees they will not be detected by the attacker. Any delay in states switching will definitely incur a risk. We also take a conservative manner in determining the timers $T_1(i)$, $T_2(i)$ and $T_3(i)$,

$$T_1(i) = T_3(i) = max\{T, T + (1 - \frac{u(i)}{U_{ref}}) * T\} \quad (7)$$

$$T_2(i) = 2 * T_1(i), \quad (8)$$

in which, T is an adjustable parameter. We let the sensors switch back to sensing/sending state at different time. Otherwise, it is risky when the attack is still nearby. Ideally, the value of T depends on the attacker information such as speed, memory content and sensing ability. However, the sensors have no knowledge about these information, so they need to be conservative in estimating the value of T, which can be based on the knowledge of maximum speed and sensing ability. We will show the sensitivity of our defense protocol's effectiveness to T and give guidelines for choosing a reasonable T in Section IV.

E. Discussions

In our defense protocol, we assume the sensors can detect the attacker and their detection range is smaller than that of the attacker. One may argue that sensors may not always be able to detect the attacker. We would like to point out that even if the sensor does not have the ability to detect the attacker remotely, it may still be able to send an AN message just before being destroyed. In the case when the destroyed sensor is not able to send out AN message before destruction, its neighbors can use some sensor fault detection methods [10], [11] to detect the destroyed node and send out AN message for it.

In our protocol, we also assume that the sensors do not have a priori knowledge about the attacker information. However, some attacker information such as v, R_{as} and R_{ps} may be obtained either by run-time measurements or off-line knowledge. In case these information are known or a good estimation like upper bound is available for the sensors, we can even obtain optimal utility threshold and optimal timers, which is one of our future work.

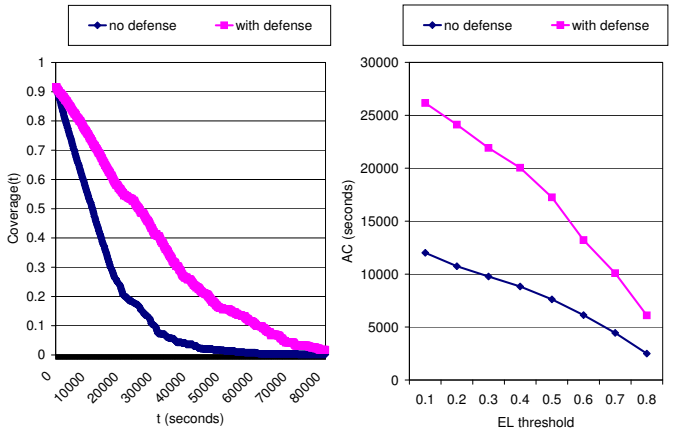
As the attacker moves in the sensing field, all the sensors along the path of the attacker will be destroyed, which may partition the network. Our defense protocol alleviates this problem by protecting some sensors along the path of the attacker from detection, thus maintaining the connectivity of the network. Besides, we may adapt our defense protocol further to maintain the connectivity. For example, the sensors whose destruction may cause network partitioning will be given extra protection and will not be chosen as *sacrificial nodes*. The study of the impact of network partitioning caused by the physical attack and the corresponding countermeasures are parts of our on-going work.

IV. Performance Evaluations

In this section, we report our performance evaluations of the impacts of search-based physical attacks on sensor networks and the effectiveness of our defense protocol in resisting attacks. Besides, we will show the sensitivity of the performance improvement to various network parameters, attacker parameters and defense parameters. As mentioned in Section III, our main performance metric here is the accumulative coverage. The search-based attack model and our defense protocol used are the ones described earlier in Sections II and III respectively.

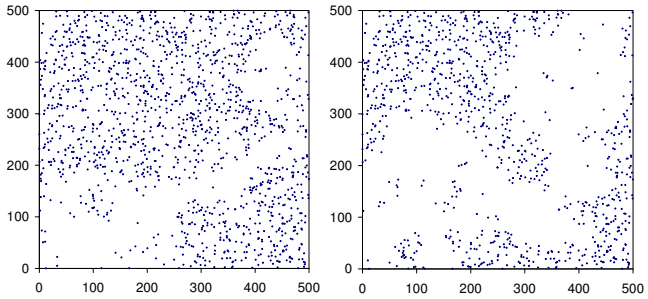
In our simulation, the sensor network area is a 500 m $\times$ 500 m square, in which 2000 sensors are randomly uniformly distributed. The active signals are generated following a constant frequency f, which may collide with the randomly generated AN/SN messages. If a collision happens, both packets are lost. The following are the default values of specific parameters used in the simulations, unless otherwise stated. $\alpha = 0.5$; $f = \frac{1}{60\ seconds}$; $R_{noti} = 20\ meters$; $R_{as} = 20\ meters$; $R_{ps} = 5\ meters$; $R_a = 0.1\ meter$; $R_s = 10\ meters$; maximum sweeping area radius [6]$= 1\ meter$; $v = 1\ meter/second$; $M = 2000$; $U_{th} = 0.7 * U_{ref}$; $\Delta t = 0.1\ second$; $T = 20\ seconds$. Our main performance metric is the Accumulative Coverage (AC). Each point of data in the figures is the average value of the results from multiple simulations with different network topologies.

A. Performance Comparisons



(a) Instant coverage. (b) AC improvement.

Fig. 4. Performance improvement of the defense protocol.



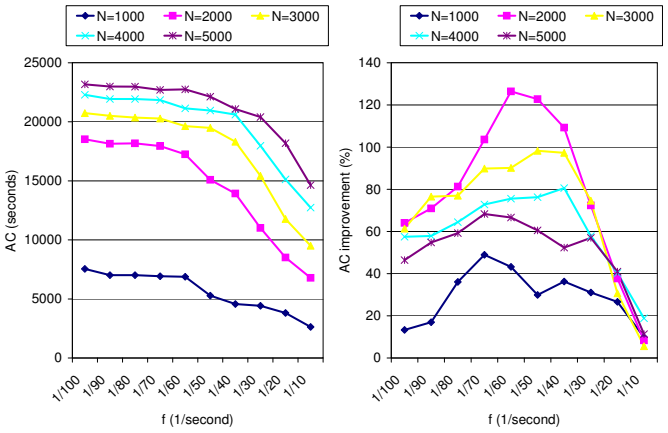
(a) with defense (b) without defense

Fig. 5. Instant alive sensors at time 10000 seconds.

In fig. 4, we show the instant coverage loss of the sensor network during attack in order to demonstrate how our defense protocol improves AC over time. As discussed in Section III, AC is the integration of $coverage(t)$, the instant coverage of the sensor network. We protect the sensor networks against search-based physical attacks by balancing short term performance (coverage(t)), and the long term performance (AC). Fig. 4 includes the scenarios with our defense protocol and without defense. In fig. 4 (a), $coverage(t)$ in the $Y - axis$ is in the time domain under search-based physical attacks. The data show that only at the very beginning (about 100 seconds), the coverage without defense is slightly larger than that with defense. However, when time goes on, the coverage value of the former drops much faster than that of the latter. The reason is, the defense mechanism forces some sensors to sleep temporarily, which decreases the short term temporary coverage. But this prevents these sensors from being detected and hence destroyed. Consequently, these sensors can have longer lifetimes and provide higher sensing coverage for longer time, and as a result long term coverage is increased.

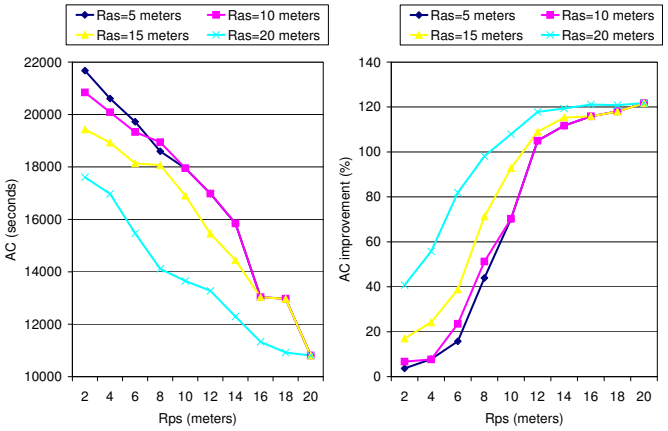
The improvement in terms of AC is more clearly shown in fig. 4 (b), in which we campare the AC with and without defense protocol under various network coverage threshold α, ranging from 0.1 to 0.8. We can see that AC decreases when α increases in both sinarios, following a close to linear pattern. However, the AC with our defense protocol consistantly outperm that without defense, ranging from 100% to 150% in

[6]We assume the sweeping area is a circle, the radius of which is proportional to the distance between the attacker and the detected sensor.



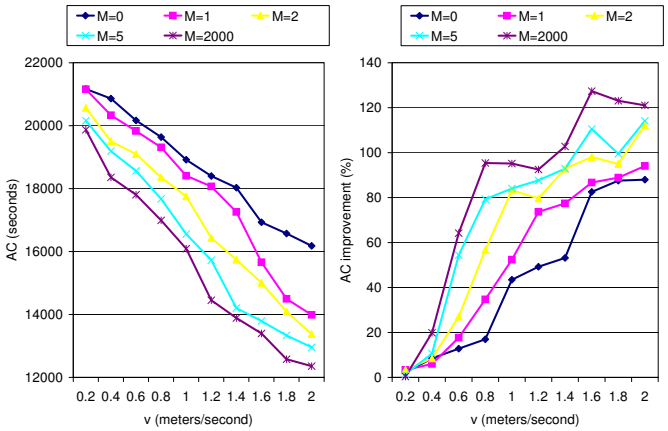
(a) Performance of defense. (b) Performance improvement.

Fig. 6. Performance comparisons under different network parameters.



(a) Performance of defense. (b) Performance improvement.

Fig. 7. Performance comparisons under different attack parameters.



(a) Performance of defense. (b) Performance improvement.

Fig. 8. Performance comparisons under different attack parameters.

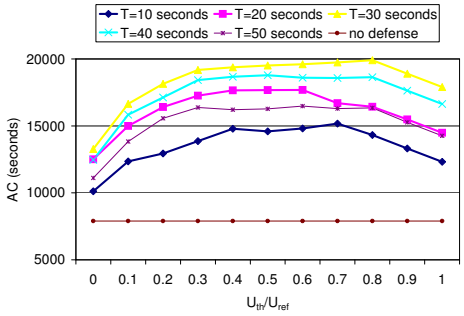


Fig. 9. Performance comparisons under different defense parameters.

terms of AC improvement. In the following subsections, we will further investigate the performance improvement of our defenese protocol under various network parameters, attack parameters and defense parameters.

In fig. 5, we show the instanct alive sensors in the network (at time 10000 seconds) under search-based physical attacks in order to demonstrate another benefit our defense protocol brings about. It is clear that with our defense protocol, not only the number of alive nodes is significantly increased, but also those alive sensors are distributed more uniformly in the sense field. Without defense, the attacker can destroy all alive sensors along its path, leaving many large holes in the field, while with our defense protocol, some sensors can protect themselves and help to decrease the number and the size of the uncovered holes. One important result is that the alive sensors in fig. 5 (b) are generally connected, while the large holes in fig. 5 (a) partition the network into a few connected components. We want to emphasize that usually at most of these connected components, which is conneted with the sigle base station, is functional and the data from other connected components cannot be delivered to the base station. If the only functional component is not large enough, the network is ragarded as unfunctional. Our current metric AC does not consider the network partitioning, which is one of our future work. We expect that the performance improvement of our defense protocol will be even more significant under the new metric. In this paper, we only show one instant case because of space limitation. We observe similar pattern under various parameters and other time instance.

B. Sensitivity of Performance Improvement under Network Parameters

In the following, we will investigate the sensitivity of performance improvement under two key network parameters, which are network density and active signal frequency. We will change the number of sensors N while fixing the size of the field. The number of sensors varies between 1000 and 5000. It is easy to calculate that the corresponding average number of neighbors for a single sensor varies between 5, which corresponds to a sparse network, and 25, which is a dense network. The active signal frequency f ranges from one packet per 10 seconds to one packet per 100 seconds, which captures the sampling rate of most sensor network applications.

In fig. 6 (a), we show the AC achieved by our defense protocol under different combinations of sensor number N and active signal frequency f. It is obvious that AC decreases when f increases. What's interesting is that there exists a threshold of f at around one packet per 60 seconds. When f is smaller than the threshold, AC decreases slightly with the increase of f. However, beyond the threshold, AC decreases sharply with the increase of f. This is because for small f, sensors send out active signal infrequently, so most sensors are detected by passive signals. In this case, decreasing f brings little extra benefit. Contrarily, when f is above the threshold, most sensors are detected by active signals due to the high frequence of packing sending and the fact that R_{as} is usually larger than R_{ps}. In this case, active signals dominates the effetiveness of attack and increasing f will significantly affects the performance. The existance of the threshold can help the network designer choose a reasonable f to make a good tradeoff between security and network performance. While a small f helps to improve security, it may decrease the network performance by introducing a long delay between sensor sampling/event detection and sending our active signal. A f a little smaller than the threshold can achieve reasonable security while introducing little compromise to the network performance.

Interestly, we observe similar effect of N on AC. First, AC increases with N due to the redundancy in more dense network. With same number of destroyed sensors, a more dense network can still maintain high coverage compared with a sparce network, thus increasing AC. More interestly, we also observe a threshold for N beyond which the extra benefit brought by dense network diminishes. This is because when N is large enough, most area of the sensor field is covered by multiple sensors. As long as at least one sensor is active, that area is considered covered in the current metric AC. Recall that our defense protocol can make the alive sensors more uniformly distributed, adding more redundancy may not help much. Same as above, the existance of the threshold can also help to choose a reasonable network density to make a good tradeoff between security and network performance. When we are only interested in 1-coverage, the extra benefit brought by adding more sensors diminishes beyond certain threshold. We

plan to consider the redundancy in the metric as our future work and we expect the performance improvement of our defense protocol will be more significantly under the new metric.

In fig. 6 (b), we show the AC improvement percentage of our defense protocol under different combinations of N and f. It is clear that AC improvement percentage increases first with f and then decreases. When f is small, the dominating factor is passive signal, in which case our defense protocol helps to improve AC mainly by avoiding sensors being detected by passive signals. When f increases, active signal becomes comparably important with passive signal. The fact that our defense protocol protects sensors from being detected by both active signals and passive signals brings about double effect, thus the AC improvement percentage increases sharply. However, when f is large enough so that active signal becomes the dominating factor, many sensors send out active signal and are detected by the attcker before receiving any notification and conducting state switching accordingly. In the extreme case, the sensors send out active signals so frequencly such that there is no way to nofity them the existance of attacker in time. Thus the AC improvement percentage decreases sharply for large f.

Similarly, we observe that AC improvement percentage increases first with N and then decreases. When N is small, most sensors have very few neighbors due to the randomly uniform distribution. Therefore, the utility value will be relatively small, which means the benefit for one notification message or one *sacrificial node* is relatively small. In this case, the AC improvement percentage is small and will increase with N. However, when N is beyond certain threshold, the network is so dense such that the attack can keep detecting new sensors in most of the time by either signal via the help of large memory size and large number of potentially detectable sensors in its detection range. In this case, the AC improvement percentage decreases with N. However, we observe that except sparce network and large f, the AC improvement percentage is still above 50%.

The above discussion shows that there exists a threshold for both network density and active signal frequency, the knowledge of which helps the network designer in choosing reasonable N and f for good tradeoff between security and network performance.

C. Sensitivity of Performance Improvement under Attack Parameters

In the following, we will investigate the sensitivity of performance improvement under four key attack parameters, which are R_{as}, R_{ps}, v and M. We will investigate different combinations of R_{as} and R_{ps} between 0 and maximum communication range R_{comm}. We will vary v between 0 and 2 $meters/second$, which covers the range of most robots and human beings. For M, we will investigate all possible values, ranging from 0 to N.

In fig. 7 (a), we show the AC achieved by our defense protocol under different combinations of R_{as} and R_{ps}. It is clear that AC decreases with the increase of either R_{as} or R_{ps} due to the increasing detecting ability of the attacker. Note that the curve for $R_{as} = 5\ meters$ and $R_{as} = 10\ meters$ merges when R_{ps} is larger than $10\ meters$. This is because in our attack model, the passive signals have higher priority than active signals because of the higher detection accuracy. When R_{ps} becomes larger than R_{as}, all sensors are detected by passive signals and the change of R_{as} has no impact on AC. Similarly, part of some curves emerge when R_{ps} is larger than $15\ meters$ and when R_{ps} is equal to $20\ meters$.

In fig. 7 (b), we show the AC improvement percentage of our defense protocol under different combinations of R_{as} and R_{ps}. We can see that the AC improvement percentage increases with R_{ps} and saturates when R_{ps} approach R_{noti}. The reason is that larger R_{ps} implies stronger attacker and it results in significant performance degradation for the sinario without defense. Our defense protocol helps to improve the AC, especially for strong attacker, rendering the increase of AC improvement percentage. However, for R_{ps} close to R_{noti}, most sensors receiving attack notification messages are already detected by attacker already. The benefit brought about by our defense protocol comes mainly from the *sacrificial nodes* and their sacrificial node notification messages, thus the AC improvement percentage saturates.

Similary to the above, the AC improvement percentage also increases with R_{as}, demonstrating that our defense protocol becomes more effective for strong attacker. However, AC improvement percentage increases more significantly when R_{as} is large, which is different from the case for R_{ps}. This is because both the notification message and corresponding state switching in our defense protocol help to prevent sensors from detected from active signals. The benefit is more obvious for large R_{as} since the performance degrades significantly in this case when no defense is applied. Similarly, part of some curves merge due to the same reason mentioned above.

In fig. 8 (a), we show the AC achieved by our defense protocol under various of v and M. It is clear that AC decreases with the increases of both v and M. However, the sensitivity of AC to v and M are quite different. We can see that AC decreases with the increases of v, following a close to linear pattern. The reason why a large v significantly decreases AC is because a fast attacker can move along a larger area, thus detecting and destroying more sensors. Interestly, the AC is not so sensitive to M. Only the initial increase of M helps the attacker to decrease AC much, beyond which there is little help for the attacker. This is because in most situations in our defense protocol, most sensors around the attacker are in sensing/sleeping states due to the notification and conservative state switching. Therefore, the attacker cannot detect many sensors in most of the time. Although a few memory size helps the attacker to store the information of multiple detected sensors, a large memory size does not help much since it is almost never used.

In fig. 8 (b), we show the AC improvement percentage of our defense protocol under various of v and M. It is clear that the AC improvement percentage increases sharply with the increase of v. This is because for small v, some sensors may switch back to sensing/sending state before the attacker leaves, thus giving a relatively low AC improvement

percentage. As mentioned in the previous section, this problem can be allieviated if the sensors are able to detect the speed of the attacker and adjust the state switching timers accordingly. Speed detection by sensors are easy to achieve via multiple sampling, which is a common assumption in many papers []. When v increases, our conservative state switching ensures that sensors will not switch back too early, thus increasing the AC improvement percentage. On the other hand, the AC improvement percentage increases with M, demonstrating that our defense protocol can allieviate the effect the large memory size and especially effective for strong attacker. Also the AC improvement percentage saturates for large M due to the same reason discussed above.

D. Sensitivity of Performance Improvement under Defense Parameters

In the following, we will investigate the sensitivity of performance improvement under two key defense parameters, which are U_{th} and T. Since the performance for the scenario when there is no defense does not depend on defense parameters, actually there is no defense at all, the AC achieved by our defense protocol has the same pattern as the AC improvement percentage. Thus we show the AC of both scenarios in one figure.

In fig. 9, we can see that the AC keeps constant for the scenario when there is no defense due to the reason above. When defense is applied, AC increases first with U_{th} at first and decreases when U_{th} approaches U_{max}. This is because for small U_{th}, there will be many sensors becoming *sacrificial nodes*, which is just not necessary. What's more, too many *sacrificial nodes* renders themselves being the potential targets for the attackers, thus decreasing the AC. In this case, the AC improvement comes only from the state switching corresponding to the sacrificial node notification messages. On the other hand, when U_{th} is close to U_{max}, too few sensors become *sacrificial nodes*, which makes many sensors outside the attack notification range dangerous, thus decreasing the AC. However, the AC in this case is still much better than that when U_{th} is too small because the AC improvement in this case comes from the state switching corresponding to both types of notification messages. The benefit of one attack notification messages usually is more than that of one sacrificial node notification messages because more sensors are notified and those sensors are more likely to be detected if they do not schedule a state switching. While it is hard, if not impossible at all, to derive the optimal utility threshold, the good news is that the AC keeps in a relatively high value for a wide range of U_{th}. In this range, the benefit of large U_{th}, which is fewer number of *sacrificial nodes* potentially being detected is allieviated by the loss in terms of number of sensors protected outside of the attack notification area. In practice, we can first choose a U_{th} close to half U_{max} and adjust the value when time goes on, during which we may be able to gain some knowledge about the attacker parameters. The way to incrementally gain and propogate attacker parameters and the internal relation between attack parameters and optimal U_{th} are part of our future work.

Similarly, we observe that AC increases first with T at first and decreases after some threshold. The reason is that when T is small, many sensors switch back to sensing/sending state before attacker leaves. Thus increasing T in this case helps to improve AC significantly. However, a too large T is not only unnecessary because the attacker has left long before the timers expire, but also affects the network performance in terms of coverage and packet throughput/delay. In this paper, we use AC to capture the performance in terms of network coverage. The study of the performance in terms of packet throughput/delay is our future work. Same as above, an optimal T is hard to derive without knowledge of attacker information. In practice, we can apply conservative timers in the beginning and adjust them when more attacker information are obtained. The relation between attack parameters and optimal T is our future work.

Discussioin: The parameters we have chosen for covers most of the practical network scenario, attacker technology and defense mechanism. It is clearly shown that the performance improvement by our defense protocol is between 50% and 150% under most normal situations, which demonstrates the validity and effectiveness of our defense protocol in protecting sensor networks performance even under search-based physical-attacks, further highlighting the significance of our study in this paper.

V. Related Work

Security in WSNs is a broad area. We highlight work most related to our study here. A good overview of current status in security and research issues is presented in [1]. Some of the security concerns include resilient routing, secure communication, and electronic and physical attacks. In [4], a survey on sensor network routing protocol vulnerabilities and defense schemes against several electronic attacks are explored. Two of these attacks are the Sybil attack [2] and the wormhole attack [3]. In [5], the authors further analyze the Sybil attack and show that it has several variants that affect data aggregation, voting, misbehavior. They also develop effective defense mechanisms against these different attack variants. In [6], Hu et al. investigate the wormhole attack and propose packet leashes to prevent an attacker from maliciously tunneling packets to different areas in a WSN. Taking another approach to routing in security, Deng et al. propose INSENS, intrusion tolerant routing that detects malicious sensors and routes around them [12]. Some of the concepts in [12] were taken from [7] which provides two security protocols, SNEP and μTESLA. These protocols insure data confidentiality, authentication, freshness and authenticated broadcast in severely resource constrained environments like WSNs, and provide defense to sybil attack, wormhole attack, eavesdrop attack [13], [1], [14], spoof, reply and message alter attack [4], [15].

In [16], the authors propose a framework for secure information aggregation in large sensor networks. In this framework certain nodes in the sensor network, called aggregators, help aggregating information requested by a query, which substantially reduces the communication overhead. By constructing efficient random sampling mechanisms and interactive proofs,

the proposed framework enables the user to verify that the answer given by the aggregator is a good approximation of the true value even when the aggregator and a fraction of the sensor nodes are corrupted.

For secure key management, [17] presents a key-management scheme based on probabilistic key sharing in WSNs. Jolly et al. develop a key management protocol for multi-tiered wireless sensor networks in [18]. Most recently, [13] solves the key management problem using *a priori* sensor network deployment information.

In [19], attackers perform traffic analysis on the messages transmitted to the base station to determine its location. A host of attacks can now be orchestrated if the base station can be determined accurately, including jamming attacks [8], eavesdropping attacks, Sybil attacks etc. In [19] and [20], protecting the base station is discussed.

Denial-of-service (DoS) attacks are another key area of vulnerability and research in WSNs. Wood and Stankovic study the threat at different layers in the network [8]. They also present design time factors that, if taken into consideration, reduces network vulnerability to DoS attacks. In [21], they further develop the radio-frequency jamming DoS attack and present a technique to route around the jammed area.

In some cases, attackers can compromise sensors with malicious intent. For instance, attackers can extract cryptographic secrets, tamper with the associated circuitry, modify programming in the sensors, or replace them with malicious sensors under the control of the attacker etc. To protect against tampering with the sensors, one defense involves tamper-proofing the node's physical package [8]. Another class of work like [22] focuses on building tamper-resistant hardware in order to make the actual data and memory contents on the sensor chip inaccessible to attackers.

Physical attacks are different from a host of sensor network attacks proposed in the literature. Physical attacks destroy sensors permanently. The losses are irreversible, unlike many other attacks, where the sensors are only compromised and hence are recoverable. In a broad spectrum of physical attacks, particularly in search-based attacks, it is very likely that the attacker physically resides in the network to detect and destroy sensors. While the attacker being in the network is dangerous, it nevertheless provides us with new opportunities for detection of the attacker and defense against them, which we have exploited in this paper. In physical attacks, attackers search for sensors by means of signal detection. This again raises several new and interesting issues, one of them being that the detection process itself can be exploited by sensor nodes as proposed in this paper. In a prior work, we have identified and modeled blind physical attacks [9]. In [9], we studied the issue of deployment of sensors in a sensor network to meet lifetime requirement under blind attacks. Our focus in this paper is search-based physical attacks, which is quite different from blind attacks as mentioned in Section II.

VI. Conclusions

In this paper we addressed the issue of search-based physical attacks in sensor networks and their defense. Specifically, we first identified critical features of search-based physical attacks and modeled a representative instance of search-based physical attacks. We studied performance impacts based on a novel metric that we defined, namely the Accumulative Coverage (AC). The accumulative coverage effectively captures both coverage and lifetime. We then proposed an effective defense protocol in order to defend sensor networks against search-based physical attacks. The core principle of our defense is to trade short term local coverage for long term global coverage through the *sacrificial node*-assisted attack notification and states switching of sensors. Our performance data demonstrated that search-based physical attacks significantly reduce accumulative coverage. However, with our defense protocol, losses in accumulative coverage are reduced significantly even under intense search-based physical attacks.

To the best of our knowledge, ours is the first work that identifies the problem, models, and defense of search-based physical attacks. We however believe that this is just an important first step in this regard. There are other open issues in this subject. Our current on-going is focusing on modeling other variants of physical attacks, and exploring defense approaches to counter them. We are specifically focusing on studying multiple physical attackers, and the combinations of physical attacks with other attacks proposed already in the literature.

References

[1] Adrian Perrig, John Stankovic, and David Wagner, "Security in wireless sensor networks," in *Communications of the ACM, Vol 47, No. 6*, June 2004, pp. 53–75.

[2] John R. Douceur, "The sybil attack," in *Proceedings of 1st International Workshop on Peer-to-Peer Systems*, March 2002.

[3] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, "Wormhole detection in wireless ad hoc networks," in *Tech. Rep. TR01-384, Department of Computer Science, Rice University*, June 2002.

[4] Chris Karlof and David Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proceedings of 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[5] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig, "The sybil attack in sensor networks: Analysis and defenses," in *Proceedings of 3rd International Symposium on Information Processing in Sensor Networks*, April 2004.

[6] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hod networks," in *Proceedings of 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, April 2003.

[7] Adrian Perrig, Robert Szewczyk, J.D. Tygar, Victor Wen, and David E. Culler, "Spins: Security protocols for sensor networks," in *Proceedings of 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, July 2001, pp. 188–199.

[8] Anthony D. Wood and John A. Stankovic, "Denial of service in sensor networks," in *IEEE Computer*, October 2002, pp. 54–62.

[9] Xun Wang, Wenjun Gu, Sriram Challeppan, Kurt Schoseck, and Dong Xuan, "Lifetime optimization of sensor networks under physical attacks," to appear in Proceedings of IEEE Internationl Conference on Communications, May 2005.

[10] K.Xing M.Ding, D.Chen and X.Cheng, "Localized fault-tolerant event boundary detection in sensor networks," to appear in IEEE INFOCOM, 2005.

[11] K.Lai S.Marti, T.J.Giuli and M.Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Mobicom '00*, August 2000.

[12] Jing Deng, Richard Han, and Shivakant Mishra, "Insens: Intrusion-tolerant routing in wireless sensor networks," in *University of Colorado, Department of Computer Science Technical Report CU-CS-9393-02*, 2002.

[13] Wenliang Du, Jing Deng, Yungshiang S. Han, Shigang Chen, and Pramod K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, March 2004.

[14] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia, "Leap: Efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, October 2003.

[15] S. Slijepcevic, V. Tsiatsis, S. Zimbeck, M.B. Srivastava, and M. Potkonjak, "On communication security in wireless ad-hoc sensor networks," in *11th IEEE Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, June 2002, pp. 139–144.

[16] Bartosz Przydatek, Dawn Song, and Adrian Perrig, "Sia: Secure information aggregation in sensor networks," in *ACM SenSys (Conference on Embedded Networked Sensor Systems)*, November 2003.

[17] Laurent Eschenauer and Virgil D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communication Security*, November 2002, pp. 41–47.

[18] Guarav Jolly, Mustaga C. Kuscu, Pallavi Kokate, and Mohamed Younis, "A low-energy key management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC '03)*, July 2003.

[19] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *Proceedings of the 2004 IEEE International Conference on Dependable Systems and Networks (DSN)*, June 2004, pp. 594–603.

[20] J. Deng, R. Han, and S. Mishra, "Enhancing base station security in wireless sensor networks," in *Technical Report CU-CS 951-03. Department of Computer Science. University of Colorado*, November 2002.

[21] Anthony D. Wood, John A. Stankovic, and Sang H. Son, "Jam: A jammed-area mapping service for sensor networks," in *Communications of the ACM, Vol 47, No. 6*, June 2004, pp. 53–75.

[22] R.J. Anderson and M.G. Kuhn, "Low cost attacks on tamper resistant devices," in *Security Protocols – Proceedings of the 5th International Workshop*, 1997, pp. 125–136.