Defending against Search-based Physical Attacks in Sensor Networks

Wenjun Gu, Xun Wang, Sriram Chellappan, Dong Xuan and Ten H. Lai

***Abstract*— In this paper we study the defense of sensor networks against *Search-based Physical Attacks*. We define search-based physical attacks as those, where an attacker detects sensors using signal detecting equipment and then physically destroys the detected sensors. In this paper, we propose a *Sacrificial Node*-assisted approach to defend 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 (AC)*, which effectively captures coverage and lifetime of the sensor networks to measure sensor network performance. Our simulation results clearly demonstrate that our defense approach can significantly decrease losses in AC even under intense search-based physical attacks.**

I. INTRODUCTION

Security has been an important research focus in Wireless Sensor Networks (WSN) recently. 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]. It is widely accepted that viability of sensor network applications in the future is closely contingent on the security of the networks.

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 due to the small form factor of sensors, and the unattended and distributed nature of their deployment. Physical attacks are relatively simple to launch and *destructive*. 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 properties (topology, routing structure etc). As such, a wide spectrum of impacts may result due to physical attacks, and when left unaddressed, physical attacks can destroy the entire sensor network mission.

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 by means of heat, physical force and other circuit tampering techniques. Another class of attacks is one, where the attacker attacks the sensor network using brute force approaches like bombs, tanks, grenades etc (which we studied in [8]). However, such brute force attacks will result in casualties to the deployment field. The attacker will prefer to conduct search-based physical attacks, when it wishes to destroy sensors, while still preserving the deployment fields without casualties. Such fields may include airports, oil-fields, battlefields etc that are of interest to the attacker.

In this paper, we first define a representative search-based physical attack model. In our attack model, the attacker continuously detects 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. A *sacrificial node* is one which detects the attacker to save other sensors from destruction at the risk of it being detected and physically destroyed by the attacker. The existence of *sacrificial nodes* compensates the weakness of the sensors' ability to detect the attacker by extending the area in which sensors are aware of the proximity of the attacker. The core principle of our *sacrificial node*-assisted defense protocol 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 clearly demonstrate that our defense approach can significantly improve the performance of sensor network even under intense attacks.

II. SEARCH-BASED PHYSICAL ATTACKS

A. Sensor Detection Mechanism

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. In our model, the attacker cannot visually detect sensors. *Active signals* on the other hand include communication messages, beacons, query messages that are part of normal network communications. These two signal types are quite different from the perspective of attacker's detection. Passive signals propagate small ranges, and the detection of them enables

Wenjun Gu, Xun Wang, Sriram Chellappan, Dong Xuan and Ten H. Lai 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.

This work was partially supported by NSF under grant No. ACI-0329155. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

the attacker to accurately detect the location of their source (the sensor emitting the passive signals). Active signals can propagate longer distance, but the attacker can only isolate the location of the source of an active signal within an area. We denote this area as the *sweeping area*. Obviously, closer the detected sensor (stronger active signal detected) is, more accurate is the isolation, and smaller is the sweeping area. We use R_{ps} and R_{as} to denote the maximal distances from where the attacker can detect passive and active signals respectively. Thus, $R_{ps} < R_{as}$. In our model, if the attacker detects multiple sensors, it can store the locations of multiple sensors in memory available to it. While the attacker can detect multiple sensors, the *target* denotes the particular sensor that the attacker currently proceeds to destroy.

The ability of the attacker to detect a sensor also depends on the state of the sensor. A sensor is *Destroyed* 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, *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 and hence cannot be detected by the attacker (even if minute signals are emitted while sleeping, we assume they are imperceptible to 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 via the signals emitted by the sensor. A sensor can instantaneously switch among these three states at will as long as it is alive.

B. The Search-based Physical Attack Model

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 sensor to destroy. Thus, 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. In our attack model, 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 1 Search-based physical attack model

```
Initialization: $Target \leftarrow \Phi$; $Mem \leftarrow \Phi$;
while the attacker is alive do
  switch type of event
  case detect a sensor $s$ through passive signal:
    $Target = \Phi$: $Target \leftarrow s$; $Target.type \leftarrow passive$;
      $Target.location \leftarrow Location\ of\ s$;
    $Target \neq \Phi$ AND $Target.type = passive$:
      Add $s$ to $Mem$;
    $Target \neq \Phi$ AND $Target.type = active$:
      Add $Target$ to $Mem$; $Target \leftarrow s$;
      $Target.type \leftarrow passive$;
      $Target.location \leftarrow Location\ of\ s$;
  case detect a sensor $s$ through active signal:
    $Target = \Phi$: $Target \leftarrow s$;
      $Target.type \leftarrow active$;
      $Target.location \leftarrow Sweeping\ area\ of\ s$;
    $Target \neq \Phi$: Add $s$ to $Mem$;
  case reach $Target.location$:
    $Target.type = passive$: Directly destroy $Target$;
      $Target \leftarrow Remove(Mem)$;
    $Target.type = active$: Sweep the sweeping area of
      $Target$; $Target \leftarrow Remove(Mem)$;
  default:
    Whenever $Target \neq \Phi$, walk towards $Target.location$,
      otherwise perform a random straight line walk;
  endswitch
end while
```

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 is a passive signal, the attacker first estimates 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 is an active signal, the attacker identifies the sweeping area and put it in memory. 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 sensors/sweeping areas can be put into memory for future targets. We denote the memory size as M.

Once the attacker reaches the target, the attacker will destroy it. If the target was detected by passive signal, the attacker will destroy the sensor directly. If only active signal was used for detecting the target, the attacker *sweeps* the sweeping area, thereby destroying any sensor in that area. Now, if the attacker has sensors in memory, it will pick the closest sensor detected by a passive signal as the next target if 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

Our search-based attack model presented here is quite representative. The philosophy of the attacker is to reach for and destroy closest sensors first. Our model can be extended to represent a wide spectrum of physical attacks. If there is no search process, the attacker can just use random sweeping to destroy sensors. This is similar to brute force attacks that we modeled in [8]. Another special case is for the attacker to destroy only specific sensors among many detected sensors. Such sensors can be cluster heads, data aggregators etc. In some cases, the attacker can aim to destroy the functionality of the sensor network by partitioning the sensor network. Such an attack may be hard to conduct, as the attacker needs a priori knowledge of the topology, communication pattern and range of sensors. In our current model, we assume there is only one attacker. However, it can be easily extended to multiple attackers if there is no cooperation among the attackers. Modeling multiple attackers with cooperation among them is part of our ongoing work. In this paper, we assume the attacker does not have the capability to reach or destroy the base station.

III. Defending against Search-based Physical Attacks

A. Design Rationale

The primary success criteria of the attacker in conducting search-based physical attacks are the coverage loss as a result of the destroyed sensors. Thus, the goal of our defense is to maintain network performance in terms of coverage under attacks. In our defense protocol, we assume that the active sensors can detect the attacker and notify its neighbors before being destroyed. The detection can be achieved by detecting signals emitted by the attacker (motion, electromagnetic etc) [9]. However, the detecting ability of the sensors is less powerful than that of the attacker. Therefore, the active sensors may only detect the attacker after they have been detected by the attacker. In this paper, we do not assume that the sensors have any knowledge about the attacker, including its speed v, memory size M, signal detection ranges R_{as} and R_{ps}.

Our defense objective is to maximize network performance in terms of coverage under search-based physical attacks. We propose a *sacrificial node*-assisted approach, in which some sensors in the detection range of the attacker choose to stay in sending state even if they are aware of the proximity of the attacker. These sensors could have switched to sensing/sleeping states to protect themselves from being detected, but they stay in the sending state to sacrifice themselves for other sensors.

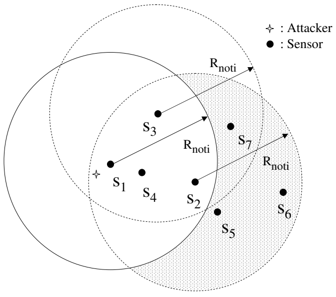

Fig. 1. An example for defense protocol description.

In Fig. 1, sensor s_1 detects the attacker and notifies its neighbors, including sensors s_2, s_3 and s_4, but sensors s_5, s_6 and s_7 are not aware of the proximity of the attacker. If sensors s_2, s_3 and s_4 switch to sensing/sleeping states and the attacker chooses to move to the right after it destroys sensor s_1, sensors s_5, s_6 and s_7 are at risk of detection. In this situation, it is important for sensor s_2 to stay in sending state so that it can notify sensors s_5, s_6 and s_7 before they are detected. Sensor s_2 could have protected itself by switching to sensing/sleeping state, however its sacrifice helps to protect many other sensors, especially when the density in the shaded area is relatively high. We call sensor s_2 a *sacrificial node*. The challenge is how sensor s_2 can decide whether it should be a *sacrificial node*, which will be described in detail in Section III-C.

Our *sacrificial node*-assisted approach helps to improve the performance of the network under search-based physical attacks by extending the area in which sensors are aware of the proximity of the attacker. The existence of *sacrificial nodes* compensates the weakness of the sensors' ability to detect the attacker. We trade a few *sacrificial nodes* for more sensors, which is a core principle of our defense protocol. Besides, our approach is localized in that the defense only involves the sensors in the local area around the attacker and each sensor's behavior is based on its local information, which makes our approach scalable and efficient.

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}	AN/SN message transmission range
R_{as}	Active signal detection range
R_{ps}	Passive signal detection range
R_a	Range within which sensor can detect attacker
R_s	Sensor's sensing range
v	Attacker speed
M	Attacker memory size
k_i	Set of sensors in sensor s_i's protection area
$k_i^{'}$	Subset of unprotected sensors in set 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 $u(i)$
$u^{opt}(i)$	Optimal $u(i)$
$u_j^{opt}(i)$	Optimal $u_j(i)$
U_{th}	Utility threshold
U_{ref}	Reference utility value
T	States switching timer parameter
$D(i)$	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

B. Defense Protocol

In this section, we first give a formal description of our defense protocol, followed by an example. The main 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 events, which is shown in Fig. 2. 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

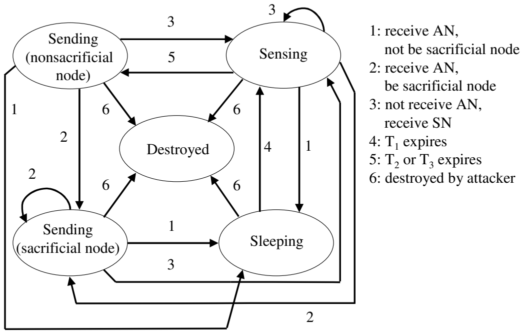
Sending (nonsacrificial node)
Sensing
Destroyed
Sending (sacrificial node)
Sleeping
3
3
5
1
2
6
6
2
4
1
6
6
1
3
2
1: receive AN, not be sacrificial node
2: receive AN, be sacrificial node
3: not receive AN, receive SN
4: T$_1$ expires
5: T$_2$ or T$_3$ expires
6: destroyed by attacker

Fig. 2. States switching and events in the defense protocol.

whether to be *sacrificial nodes* or not based on *sacrificial nodes* determination scheme (discussed later). For the recipient sensors of the AN message that decide not to be *sacrificial nodes*, called *nonsacrificial nodes*, they will calculate two timers, T_1 and T_2, and switch to sleeping state immediately (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 (event 4 and 5 in Fig. 2). For other recipient sensors of the AN message that decide to be *sacrificial nodes*, they will send out *sacrificial node* notification messages (SN messages) and stay in sending state (event 2 in Fig. 2). For the sensors that do not receive the original AN message but receive at least one corresponding SN message, they will calculate a timer T_3 and switch to sensing state immediately (event 3 in Fig. 2). These sensors will switch back to sending state as *nonsacrificial nodes* after T_3 expires (event 5 in Fig. 2). Obviously, the sensors that are destroyed by the attacker will switch to destroyed state (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 IDs of both the sensor sending out this SN message and the sensor sending out the corresponding AN message. We assume the attacker cannot create the AN/SN messages, which could be due to an existing pairwise key scheme [10]. The detailed description of *sacrificial nodes* determination 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. In the beginning, 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 s_1 and the notification area is a circle of radius R_{noti} centered at s_1. We let R_{noti} be the same as the sensor communication range. Recall that the attacker is more powerful than a sensor in terms of sensing ability. As such, s_1 will be detected by the attacker before s_1 has detected the attacker. Thus it is better for s_1 to send out AN message instead of switching to sensing/sleeping state. After sending out the AN message, s_1 will stay in sending state.

For the recipients of the AN message sent by s_1, which are s_2, s_3 and s_4, we assume s_2 and s_3 decide to be *sacrificial nodes* while 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 s_2 and s_3 send out the SN messages, they will stay in sending state as *sacrificial nodes*. The SN message of s_2 contains the global IDs of s_2 and 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. The protection area of a sensor is a circle centered at the sensor with radius R_{noti}. On the other hand, s_4 will calculate two timers, T_1 and T_2, and switch to sleeping state immediately. After T_1 and T_2 expire, s_4 will switch back to sensing and sending (as *nonsacrificial node*) states respectively.

In Fig. 1, s_5, s_6 and s_7 receive the SN message sent by s_2 and s_3, but they did not receive the corresponding AN message sent by 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 preferable for them 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. Second, they are already in the protection area of s_2. They may be notified of the approaching of the attacker by s_2 before their own passive signals are detected and then switch to sleeping state in time.

C. Sacrificial Nodes Determination

It is obvious that a sensor is more preferable to become a *sacrificial node* if it can protect more sensors. We use a utility function $u(i)$ to measure the preference for a sensor s_i being a *sacrificial node*. Our *sacrificial nodes* determination is based on this utility function calculated by each sensor.

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, s_2 is more preferable than s_4 because it can potentially protect more sensors. Thus, a simple utility function is given by,

$$u(i) = |k_i|, \tag{1}$$

in which k_i denotes the set of sensors in the protection area of sensor s_i and $|k_i|$ denotes the size of the set k_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. Selecting the second one as a *sacrificial node* brings little extra benefit. Besides, it incurs more risk since both of them become potential targets now. A reasonable modification is given by,

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

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

Furthermore, we observe that in calculating the utility function of a sensor s_i, the relative distances of the sensors in $k_i^{'}$ from sensor s_i also make difference. In Fig. 1, we assume the attacker moves to the right after it destroys s_1. Compared with s_6, s_5, which is closer to s_2, is more likely to be detected before s_2 detects the attacker and sends out an AN message. In this case, the contribution of s_5 to the utility function of s_2 should be smaller than that of s_6. Recall that the sensors have no knowledge of the sensing ranges of the attacker, therefore we weigh the contribution of s_j to $u(i)$, denoted as $u_j(i)$, by the distance between s_i and s_j, denoted by $d(i,j)$. The distance between neighboring sensors can be obtained by sensor localization schemes [11]. Thus we obtain the following utility function,

$$u(i) = \sum_{s_j \in k_i^{'}} u_j(i) = \sum_{s_j \in k_i^{'}} \frac{d(i,j)}{R_{noti}}. \tag{3}$$

In the ideal situation, assuming all sensors have full knowledge about the attacker, a sensor s_i can 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 $u_j(i)$. Theorem 1 states that the utility function in (3) 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. Before stating the Theorem, we introduce the following Lemma.

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 $u_j^{opt}(i)$.

We now state our theorem below.

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 $u^{opt}(i)$.

Theorem 1 can be proved using Lemma 1. Due to lack of space, we omit the proof here. For detailed proof, refer to [12].

In (3), if we replace $k_i^{'}$ by a set with size being the average number of neighbors for a sensor and replace the weight $\frac{d(i,j)}{R_{noti}}$ 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,

$$U_{ref} = \frac{N \pi R_{noti}^2}{S}, \tag{4}$$

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

2) Sacrificial Nodes Determination Scheme: We now describe the criterion used by a sensor to decide whether it should be a *sacrificial node* 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 issue of choosing a reasonable U_{th} in future work.

After the discussion of *sacrificial node* determination criterion, we will describe the scheme used by recipient sensors of an AN message for *sacrificial nodes* determination. 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. Other sensors, called candidate *sacrificial nodes*, will calculate a random delay and set a timer (denoted by $D(i)$). It is given by,

$$D(i) = \begin{cases} \epsilon * \Delta t & , \quad u(i) \geq U_{ref} \\ \Delta t + (1 - \frac{u(i)}{U_{ref}}) * \Delta t, & U_{th} \leq u(i) < U_{ref} \end{cases} \tag{5}$$

where 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*. 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 (3). 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. 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 to move 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. This could be 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. On the principle of being conservative, we determine the timers $T_1(i)$, $T_2(i)$ and $T_3(i)$ by,

$$T_1(i) = T_3(i) = max\{T, T + (1 - \frac{u(i)}{U_{ref}}) * T\}, \quad (6)$$

$$T_2(i) = 2 * T_1(i), \quad (7)$$

in which, T is an adjustable parameter. We let the sensors switch back to sensing/sending states at different time. Otherwise, it will incur more risk if the attacker is still nearby when the sensors switch back to sensing/sending state altogether. 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 this, 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 of the attacker. We will investigate the issue of choosing a reasonable T in future work.

E. Discussions

In our defense protocol, we assume the sensors can 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 via some hardware triggering mechanism. 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 [13], [14] to detect the destroyed sensor and send out AN message for it.

In our protocol, we do not assume that the sensors 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 is 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.

IV. Performance Evaluations

A. Performance Metric and Simulation Settings

In order to evaluate the performance of sensor networks under search-based physical attacks, we define a novel metric, 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, and effective lifetime is the time period until when the sensor network becomes nonfunctional because the coverage falls below a system required threshold α. Denoting *coverage(t)* as the network coverage at time t, and EL, as the effective lifetime, we have,

$$AC = \int_{t=0}^{EL} coverage(t)dt. \quad (8)$$

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, which 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.

In our simulation, the sensor network area is a 500 m × 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 AN/SN messages. If a collision happens, all packets involved are lost and no lost packet will be retransmitted. The following are the default values of specific parameters used in the simulations, unless otherwise stated. α = 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 [1]= 1 $meter$; $v = 1\ meter/second$; $M = 2000$; $U_{th} = 0.7 * U_{ref}$; $\Delta t = 0.01\ second$; $T = 20\ seconds$. Each point of data in the figures is the average value of the results from multiple simulations with different randomly generated network topologies.

B. Sensitivity of the Defense to Sensor Network Parameters

In the following, we investigate the sensitivity of the defense in terms of performance improvement to two key network parameters, namely sensor density and active signal frequency. We choose two standard values, 2000 and 4000, for N, the number of sensors, while the size of the sensing field is fixed. The corresponding average numbers of neighbors of a sensor are around 10 and 20 respectively, which corresponds to a relatively sparse network and a relatively dense one. The active signal frequency f ranges from one per 100 seconds to one per 10 seconds, which captures the sampling rates of most sensor network applications. We do not show the simulation results with other values of the network parameters due to space limitation. Interested readers can refer to [12]. However, the data we report here are representative.

Fig. 3 shows that AC decreases when f increases or when N decreases. When f is large, more sensors are detected, hence AC is smaller. When N is large, the coverage is large due to large sensor density and more redundancy of coverage. An interesting observation is that, there exists a threshold of f. When f is smaller than the threshold, AC decreases slowly with the increase of f. However, when f is larger than the threshold, AC decreases sharply with the increase of f. The reason why there exists a threshold of f is, when f is small, sensors send out active signals infrequently, so most sensors are detected through passive signals. In this case, increasing f does not change the fact that few sensors are detected through active signals. Contrarily, when f is above the threshold, most

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

sensors are detected by active signals due to the high frequency of active signals and the fact that R_{as} is larger than R_{ps}. In this case, active signals dominates the effectiveness of attack and increasing f will significantly increase the attack effectiveness. The existence of a threshold for f can help the network designer to choose a reasonable f to make a good tradeoff between AC and the throughput/delay of the network. While a small f helps to improve the resilience of the network and AC, it may decrease the network performance by reducing the throughput and increasing the communication delay of the network. A reasonable f should be smaller than but close to the threshold, which can achieve reasonable level of AC while introducing little compromise to the network throughput/delay. The value of the threshold depends on the attacker information. In case the attacker information is known or a good estimation like upper bound is available for the sensors, we may obtain the value of the threshold, which is one of our future work.

C. Sensitivity of the Defense to Attack Parameters

In the following, we investigate the sensitivity of the defense to two key attack parameters, which are attacker moving speed, v, and the size of the attacker memory, M. We vary v from 0 and 2 $meters/second$, which covers the range of the moving speed of most robots and human beings. For M, we will investigate two extreme cases, 0 and 2000. We let M be 2000 to represent the extreme case when no detected sensors will be ignored due to the limitation of memory size.

Fig. 4 shows that, AC decreases with the increases of both v and M. A large v can significantly decrease AC because a fast attacker can visit a larger area within a certain amount of time, thus detect and destroy more sensors. The second observation is, the trend of the decrease of AC over the increases of v follows an almost linear pattern when defense is applied, but AC decreases much more sharply with the increase of v when there is no defense. This confirms the effectiveness of our defense protocol under powerful attacks. The third observation is that the improvement of AC provided by our defense protocol is more significant for large v. This is because, when v is small, some sensors may switch to sleeping state much earlier before the attacker comes and switch back to sensing/sending state before the attacker leaves, which incurs a relatively low AC improvement. As mentioned before, this problem can be alleviated if the sensors are able to detect the speed of the attacker and adjust the states switching timers accordingly. Speed detection by sensors can be achieved via multiple samplings at different time. When v increases, our conservative states switching ensures that the sensors will not switch back too early, which increases the AC improvement.

The fourth observation is, AC is not so sensitive to M as to v. We observe that only the initial increase of M helps the attacker to decrease AC significantly. When M is larger than that, there is little extra help for the attacker. This is because, in most situations, the number of active sensors in the detection range of the attacker is limited due to the sensor density. Therefore, the attacker cannot detect many sensors in most of the time, thus larger memory is not so useful.

We do not report the sensitivity of performance to R_{as} and R_{ps} in this paper due to space limitation. Interested readers can refer to [12]. Basically, the increase of either R_{as} and R_{ps} decreases AC. Furthermore, the performance improvement is more significant under increasing R_{as} and/or R_{ps} values.

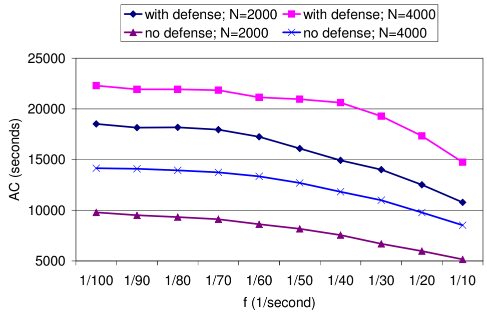

Fig. 3. Performance comparisons under different network parameters.

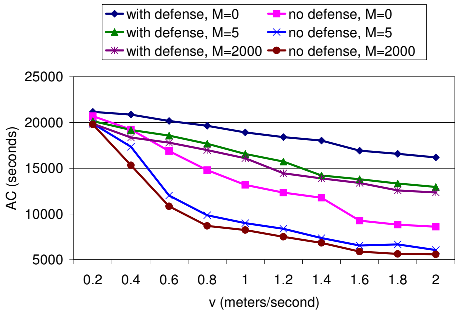

Fig. 4. Performance comparisons under different attack parameters.

V. Related Work

Security in WSNs is a broad area. A good overview of security in WSNs is presented in [1]. In [2], a survey on sensor network routing protocol vulnerabilities and defense schemes against several electronic attacks are explored. Our work is different from the above in that physical attacks destroy sensors completely, unlike many other attacks, where the sensors are only affected partially in terms of functionality. In a prior work, we have identified and modeled blind physical attacks [8]. In [8], we studied the issue of deployment of sensors in a sensor network to meet lifetime requirement under blind physical attacks. Our focus in this paper is search-based physical attacks, which is quite different from blind physical attacks.

A type of attack related to physical attacks is jamming attacks [3], [15], 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.

In some cases, attackers can compromise sensors with malicious intent. For instance, attackers can extract cryptographic secrets, replace them with malicious sensors under the control of the attacker etc. To protect against sensor tampering, one defense involves tamper-proofing the node's physical package [3]. Another class of work like [16] focuses on building tamper-resistant hardware to make the memory contents inaccessible to attackers. While the above work tries to protect sensors' physical security via improved hardware, which may not be always achievable under powerful physical attacks, we propose a defense protocol that does not assume any indestructible hardware and can alleviate the destruction of physical attacks significantly.

Li et al. [17] propose a distributed algorithm for guiding a user across a sensor network. The user can communicate with the sensors and thus avoid some danger areas in the network. Corke et al. [18] study a deployment problem in sensor network in which autonomous aerial vehicles communicate with sensors deployed and determine the gaps in connectivity, which is used for a later repair process. In these works, sensors help users, which could be human beings, robots or autonomous aerial vehicles, achieve certain goal via communication. In this paper, we are addressing a different problem, in which sensors cooperate with each other to defend against attacks via local communication.

Gui et al. [19] study the trade-off between power consumption and quality of surveillance in event tracking. In [19], sensors around a moving event notify other sensors in the neighborhood such that the sensors nearby, which are in low power surveillance state, can switch to high power tracking state in time to achieve good quality of surveillance with minimum power consumption. Constant event speed is assumed in [19]. A similar work is [20], in which sensors cooperate with each other via messages so that only the sensors around a moving event are in tracking state. In [20], the speed and moving direction of the event are assumed to be known/measurable by the sensors. While the attack notification and states switching in our defense protocol bear some similarities with the above work, we have one extra constraint, minimizing the number of messages for notification that can be detected by the attacker. The fundamental tradeoff between cooperation via messages and minimizing number of messages being detected, and that between providing sensing coverage and avoiding being detected makes our problem more challenging. Besides in our model, sensors do not posses any attacker knowledge, like speed, moving direction etc.

VI. Conclusions

In this paper we addressed the issue of search-based physical attacks in sensor networks and their defense. Specifically, we first modeled a representative instance of search-based physical attacks. We then propose a *sacrificial node*-assisted defense protocol to defend 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. We studied performance impacts based on a novel metric that we defined, namely the Accumulative Coverage (AC). Our simulation results demonstrated that our defense protocol significantly improves sensor network performance 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. Our current ongoing work is focusing on studying multiple cooperative physical attackers.

References

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

[2] C.Karlof and D.Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (WSNA)*, May 2003.

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

[4] A.Perrig, R.Szewczyk, J.D.Tygar, V.Wen, and D.E.Culler, "Spins: Security protocols for sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, July 2001.

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

[6] J.Newsome, E.Shi, D.Song, and A.Perrig, "The sybil attack in sensor networks: Analysis and defenses," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, April 2004.

[7] Y.C.Hu, A.Perrig, and D.B.Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hod networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2003.

[8] X.Wang, W.Gu, S.Challeppan, K.Schoseck, and D.Xuan, "Lifetime optimization of sensor networks under physical attacks," in *Proceedings of IEEE International Conference on Communications (ICC)*, May 2005.

[9] P.Dutta, M.Grimmer, A.Arora, S.Bibyk, and D.Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *4th International Conference on Information Processing in Sensor Networks (IPSN)*, April 2005.

[10] Z.Yu and Y.Guan, "A key pre-distribution scheme using deployment knowledge for wireless sensor networks," in *Proceedings of the 4th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, April 2005.

[11] C.Wang, L.Xiao, and J.Rong, "Sensor localization in an obstructed environment," in *Proceedings of the 1st IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2005.

[12] W.Gu, X.Wang, S.Chellappan, and D.Xuan, *Defending against Physical Attacks in Sensor Networks*, The Ohio State University, The Department of Computer Science and Engineering, Technical Report, 2005.

[13] K.Xing M.Ding, D.Chen and X.Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2005.

[14] K.Lai S.Marti, T.J.Giuli and M.Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (Mobicom)*, August 2000.

[15] A.D.Wood, J.A.Stankovic, and S.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.

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

[17] Q.Li, D.R.Michael, and R.Daniela, "Distributed algorithms for guiding navigation across a sensor network," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, September 2003.

[18] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Deployment and connectivity repair of a sensor net with a flying robot," in *Proceedings of the 9th International Symposium on Experimental Robotics (ISER)*, June 2004.

[19] C.Gui and P.Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, September 2004.

[20] Q.Huang, S.Bhattacharya, C.Lu, and G.C.Roman, "Far: Face-aware routing for mobicast in large-scale sensor networks," to appear in ACM Transactions on Sensor Networks, 2005.