DEFENDING AGAINST PHYSICAL DESTRUCTION ATTACKS ON WIRELESS SENSOR NETWORKS

Chi Zhang, Yanchao Zhang, Yuguang Fang
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611
Tel: (352)392-2746; Fax: (352)392-0044; Email: {zhangchi@, yczhang@, fang@ece.}ufl.edu

ABSTRACT

In order to defeat physical destruction attacks on wireless sensor networks, the base station (BS) need have the ability to continuously self-monitor the change of the WSN's coverage performance. In this paper, we propose a secure coverage inference protocol (SCIP) which can provide the BS an accurate and in-time measurement of the current connected coverage in an energy-efficient way, and our SCIP is resilient to attacks from external adversaries as well as to attacks from compromised nodes.

I. INTRODUCTION

The small form factor of sensors, coupled with the unattended and distributed nature of their deployment expose *wireless sensor networks* (WSNs) to *physical destruction attacks* which are more likely than in most other distributed systems. In this kind of attacks, the adversary permanently destroys sensor nodes' sensing and communication abilities by hurling real grenades/bombs or using electromagnetic burst weapons to emit a strong pulse on selected areas in the geographic region under the WSN's surveillance (called the *region of interest* or ROI). A smarter attacker even can detect and destroy sensors at key positions with stealth by moving across the ROI. Physical attacks are inevitable threats in WSNs: they are relatively simple to launch and fatal in destruction.

A position in the ROI is really under the surveillance of the WSN if and only if this position is within the sensing range of at least one sensor node connected to the *base station* (BS). We define the collection of all these positions in the ROI as the *connected coverage* (or *coverage* in short). To effectively defeat physical destruction attacks, the BS should have the ability to self-monitor the change of the WSN's coverage performance. First of all, connected coverage is one of the most important performance metrics measuring the quality of surveillance a WSN can provide, and should be an inseparable complementarity of the report about the observed events in the ROI. Secondly, the details of coverage information can help decide when and how to perform the network repair or re-deployment. More interestingly, if the information about when and where the sensors are destroyed is extracted from the continuous coverage measurement, it can provide the WSN with vital clues about an enemys intent, strategy and position, and aid users to launch counter-attacks.

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464.

However, due to the special nature of WSNs, the task of design a practical and robust scheme to continuously measure the connected coverage is not trivial at all. One significant challenge facing comes from the strict resource limitation of the sensor nodes (battery power, memory, computational ability, etc.), which highlights the requirements of energy and computational efficiency. The unattended and distributed nature of the deployment of sensor nodes without tamper-proof also expose them to node capture attacks, and constitutes another significant challenge.

To the best of our knowledge, this paper is the first work to provide an energy-efficient and compromise-tolerant scheme for the coverage measurement. Our contribution is twofold. First, we design a basic *coverage inference protocol* (CIP) based on a novel *boundary node detection scheme* (BOND), which only needs one-hop neighbors' information, and achieves energy efficiency and measurement accuracy. Second, we propose a *location-based symmetric key management protocol* (LBSK), and integrate it to our basic CIP. We show that this secured measurement scheme is resilient to attacks from external attackers as well as to attacks from a subset of compromised nodes.

II. ASSUMPTIONS AND DESIGN GOALS

A. Network and Security Assumptions

Throughout this paper, we assume that any two sensor nodes can directly exchange messages if their Euclidean distance is not greater than r_c, the *communication range*; and a position in the plane can be perfectly monitored (or covered) by a sensor node if their Euclidean distance is not greater than r_s, the *sensing range*. For simplicity, we also assume that sensor nodes are homogeneous, i.e., r_c and r_s are the same for all nodes, and $r_c = 2r_s$, which holds for most commercially available sensors such as Berkeley Motes [20]. However, it should be noted that our algorithms

are still applicable to the scenarios of $r_c > 2r_s$. We denote the square ROI of side length l as A_l, and the *border* of A_l as ∂A_l. We consider a network made of stationary sensor nodes only, and denote the sensor nodes deployed in the ROI to be $V = \{s_1, \cdots, s_i, \cdots, s_n\}$ ($s_i \in A_l$, for $1 \leq i \leq n, i \in \mathbb{N}$), where s_i represents the position of node i and n is the total number of sensor nodes, or *network size*.

It is unrealistic and uneconomical to enclose each node in tamper-resistant hardware. Thus, each sensor node represents a potential point of compromise. However, we assume that sensor nodes deployed in security-sensitive environments are usually designed to withstand break-in attacks at least for a short time interval T_{min}, which is reasonable in practice, and widely accepted in the literature [21], [22]. If a node is compromised, the adversary can acquire its key material, gain full control over the node, and easily insert its clones in the ROI. We postulate that the BS is unassailable, as is commonly assumed in the literature [21], [22]. We also assume some kinds of intrusion detection based on neighborhood monitoring, like [5], [7] are used in the WSN, so that apparent misbehaving nodes will be detected and isolated by the neighboring nodes.

B. Design Goals

The main goal of this paper is to design a *secure coverage inference protocol* (SCIP) which can provide the BS an *accurate* and *in-time* measurement of the current connected coverage in an *energy-efficient* way. Since the adversary has the incentive to eavesdrop, alter, forge and hide the coverage information, our protocol must be designed with security requirements in mind. More specifically,

Measurement Confidentiality: The procedure of the measurement should not be utilized by adversaries to figure out the coverage situation of the WSN. The compromise of nodes should only result in the revealing of the coverage information of a small, localized part of the WSN.

Measurement Integrity and Authenticity: When the measurement is proceeded in a distributed fashion, our protocol need to ensure that the data as the partial result of the measurement will not be altered by the intermediate relay nodes. When received data is determined as false, the BS should have the ability to detect the sources of those false data (i.e., non-repudiation). The effect of compromised nodes (including their clones) on the final measurement result should be minimized or localized.

III. BOND: BOUNDARY NODE DETECTION SCHEME

A connected set of nodes is said to be a *cluster* if the addition of any other node to it will break the connectedness property. Obviously, the problem of finding the boundary of connected coverage, is equivalent to detecting the boundary nodes of clusters with connections to the BS. Based on this observation, it is possible to design a distributed SCIP if we can first find a localized way to detect boundary nodes.

In this section, we present the first component of our SCIP: BOND scheme which provide an energy-efficient way for each node to locally self-detect whether it is a boundary node. we begin with several definitions:

A. Localized Voronoi Polygons

Our BOND scheme is based on two novel geometric concepts called *localized Voronoi polygon* (LVP) and *tentative LVP* (TLVP) which are nontrivial adaptations of *Voronoi polygons* (VPs) [16] from computational geometry.

We first define VPs, LVPs and TLVPs in terms of half planes. For two distinct points $s_i, s_j \in V$, the *dominance region* of s_i over s_j is defined as the set of points which are at least as close to s_i as to s_j, i.e.,

$$Dom(s_i, s_j) = \left\{ v \in \mathbb{R}^2 : \|v - s_i\| \leq \|v - s_j\| \right\}. \quad (1)$$

Obviously, $Dom(s_i, s_j)$ is a half place bounded by the perpendicular bisector of s_i and s_j, which separates all points in the plane closer to s_i than those closer to s_j.

***Definition** 1: The **VP** associated with s_i denoted by $Vor(s_i)$, is the subset of the place that lies in all the dominance regions of s_i over other points in V, namely,*

$$Vor(s_i) = \bigcap\nolimits_{s_j \in V - \{s_i\}} Dom(s_i, s_j). \quad (2)$$

*In the same way, the **LVP** denoted by $LVor(s_i)$, and the **TLVP** denoted by $TLVor(s_i)$, are defined as:*

$$LVor(s_i) = \bigcap\nolimits_{s_j \in Neig(s_i)} Dom(s_i, s_j); \quad (3)$$

$$TLVor(s_i) = \bigcap\nolimits_{s_j \in SubNeig(s_i)} Dom(s_i, s_j), \quad (4)$$

where $SubNeig(s_i)$ is a proper subset of $Neig(s_i)$, i.e., $SubNeig(s_i) \subset Neig(s_i)$.

The collection of LVPs given by

$$\mathfrak{LVor}(V) = \{LVor(s_1), \cdots, LVor(s_n)\} \quad (5)$$

is called the *localized Voronoi diagram* (LVD) generated by the node set *V*. The boundary of $LVor(s_i)$, i.e., $\partial LVor(s_i)$, may consist of line segments, half lines, or infinite lines, which are all called *local Voronoi edges*.

B. LVP-Based Boundary Node Detection

In this subsection, we present our BOND scheme for each node to detect whether it is a boundary node based on its own LVP or TLVP by taking node s_i as an example.

1) Input: Our BOND is a distributed scheme in that we only need positions of node s_i's neighbors as the input of our algorithm. We need to consider two cases based on whether the information about the border of A_l, i.e., ∂A_l, is available. In the first case when ∂A_l is unavailable at node s_i, our detection scheme is based on the construction of $LVor\,(s_i)$ (or $TLVor\,(s_i)$); in the second case when ∂A_l is available, we need to exploit this information by calculating $LVor\,(s_i) \cap A_l$ (or $TLVor\,(s_i) \cap A_l$). It can be shown that $LVor\,(s_i) \cap A_l$ must be a finite convex polygon. Thus, the second case can be transformed into the first case by introducing dummy nodes into $Neig(s_i)$. When ROI is a square field, four dummy nodes, d_1 through d_4, are introduced such that perpendicular bisectors between s_i and the dummy nodes generate the four border edges of ROI. Then we can calculate $LVor\,(s_i) \cap A_l$ by following the same procedure for calculating $LVor\,(s_i)$. Therefore, we will only discuss the first case in what follows.

2) Algorithm: Our goal is to construct the $LVor(s_i)$ (or $TLVor(s_i)$) which is sufficient for the boundary node detection with the minimal requirement on the information about s_i's neighbors. We first divide $Disk\,(s_i, r_c)$ into four (other values will also work well) quadrants. Then we construct the TLVP of s_i by using the nearest neighbors in each of the four quadrants. Without lose of generality, we denote these four nearest neighbors as s_1, s_2, s_3, and s_4. The first TLVP is calculated by

$$TLVor\,(s_i) \leftarrow \bigcap\nolimits_{j=1}^{4} Dom(s_i, s_j).$$

If all vertices of the TLVP is covered by $Disk\,(s_i, r_s)$, the procedure stops and this TLVP is saved. Otherwise, we need to find new neighbors which are the nearest to the uncovered vertices of the TLVP, add those neighbors to $SubNeig(s_i)$, and calculate the TLVP again:

$$TLVor\,(s_i) \leftarrow TLVor\,(s_i) \bigcap \left(\bigcap\nolimits_{s_j \in SubNeig(s_i), j \neq 1,2,3,4} Dom\,(s_i, s_j)\right).$$

The new vertices of the new TLVP will be checked to see whether they are covered by $Disk\,(s_i, r_s)$. This procedure continues until all the vertices of the TLVP are covered by $Disk(s_i, r_s)$ or the LVP of s_i is calculated and saved.

Note that when ∂A_l is unavailable, $LVor(s_i)$ may be infinite, which means that it is possible that we cannot find any nodes in one or more quadrants in the first step. If a quadrant contains no neighbors, we define two sectors of angle 45° which are directly adjacent to the quadrant as the assistant area, and add the nodes in this area to $SubNeig(s_i)$ first. If all the nodes in the assistant area cannot make TLVP finite, we can conclude that LVP must be infinite without need to do further calculation.

3) Output: If $LVor\,(s_i)$ is infinite, s_i must be a boundary node. If $LVor\,(s_i)$ (or the final $TLVor\,(s_i)$) is finite with all the vertices are covered by s_i, then $s_i \in IN\,(s_i)$. Otherwise, $s_i \in BN\,(s_i)$.

C. Discussion on BOND

Correctness. The correctness of BOND is proved in our previous work [19] for arbitrary node distributions.

Low Overhead. It has been shown in [19] that in general VPs cannot be computed locally. Therefore, the traditional VP-based schemes [4], [6] are not distributed and are very expensive in terms of communication overhead. Our BOND scheme is a truly localized polygon-based solution because computing $LVor\,(s_i)$ (or $TLVor\,(s_i)$) only needs one-hop information (this can be directly obtained from the Eqs. 3 and 4). Assuming that the number of trusted neighbors is k, each node can compute its own $LVor(s_i)$ with complexity smaller than $O(k)$. In addition, the computation of the $LVor(s_i)$ only involves some simple operations on polygons which can be efficiently implemented (e.g., PolyBoolean library [13]). We further simplify the detection process by constructing TLVPs first. For a densely deployed WSN, we have $LVor(s_i)$ or $TLVor(s_i) \rightarrow Vor(s_i)$, and it is well known in computational geometry that under the homogeneous *spatial Poisson point process* (SPPP), the average number of vertices of $Vor(s_i)$ is 6 [16]. Therefore, even when the node density is high, BOND on average only needs 4 to 6 nearest neighbors' information to successfully detect the boundary nodes. Moreover, when a neighbor node dies, BOND needs do nothing unless the dead node is used to construct the final $TLVor(s_i)$ or $LVor(s_i)$ in the last turn of LVP or TLVP construction. This unique property will greatly simplify the update of detection results and save precious energy of each sensor node. All these advantages cannot be achieved by other localized boundary node detection schemes in the literature, such as the perimeter-coverage checking approach [8] and the crossing-coverage checking approach [20]. For lack of space, we refer to [19] for a detailed comparison.

IV. LBSK: Location-Based Symmetric Key Management Protocol

This section presents the second component of our SCIP: LBSK, which establishes related keys that can be used to secure basic network operations and our BOND.

A. Pre-Deployment Phase

We assume that each node s_i has a unique, integer-valued and non-zero ID, denoted by ID_{s_i}. All the nodes used for initial network deployment have the IDs not greater than

ID_{max}. Prior to network deployment, we assume that a *trusted authority* (TA), e.g., the system administrator or network planner, first determines three initial keys K_I^1, K_I^2 and K_I^3, and one secure one-way function $H(\cdot)$ [18]. It then calculates *ID-based Key* (IBK for short) $IK_{s_i} = H(K_I^1|ID_{s_i})$ and *broadcast key* $BK_{s_i} = H(K_I^2|ID_{s_i})$ for each sensor node s_i, where $|$ denotes message concatenation. After that, the TA pre-loads each node with the following bootstrapping parameters:

$$\{K_I^1, K_I^2, ID_{max}, ID_{s_i}, IK_{s_i}, BK_{s_i}\}.$$

B. Neighbor Discovery Phase and PK Establishment

When it is deployed, each node i.e., s_i, first initializes a timer to fire after time T_{min}. It then tries to discover its neighbors. It broadcasts a *HelloNeig* message which contains its ID and waits for each neighbor s_j to respond with an ACK message including the ID of node s_j.

$$\begin{array}{lcl} s_i \rightarrow * & : & ID_{s_i}, HelloNeig, \mathsf{MIC}(IK_{s_i}, ID_{s_i}) \\ s_j \rightarrow s_i & : & ID_{s_j}, \mathsf{MIC}(IK_{s_j}, ID_{s_i}|ID_{s_j}) \end{array}$$

Here, $\mathsf{MIC}(k, M)$ refers to the *message integrity code* (MIC) of message M under key k. We use "$u \rightarrow v : M$" to denote a one-hop delivery of message M from u to a neighbor v and "$u \rightarrow\rightarrow v : M$" to denote a delivery that may involve multiple hops. Since node s_i knows K_I^1 in time T_{min}, it can derive IK_{s_j} from ID_{s_j} and then verify node s_j' s identity. After that, if $ID_{s_i} > ID_{s_j}$, node s_i computes its *pairwise key* (PK) with s_j, $PK_{s_i s_j}$, as $PK_{s_i s_j} = H(IK_{s_j}|ID_{s_i})$ and node s_j's broadcast key BK_{s_j}. Node s_j can also verify node s_i's identity and compute $PK_{s_i s_j}$ and BK_{s_i} in the same way. Note that if $ID_{s_j} > ID_{s_i}$, the pairwise key between s_i and s_j will be $PK_{s_j s_i} = H(IK_{s_i}|ID_{s_j})$. After this neighbor discovery phase, each node will have a list of authenticated neighboring nodes with corresponding PKs.

C. Localization Phase and LBK Establishment

There are many methods to localize each node, i.e., furnishing each node with its geographic location. We consider the following two localization techniques, which accordingly differ in their ways of generating LBKs for individual nodes.

1) Range-free localization: In this approach, we assume that there are some special nodes called anchors knowing their own locations. All the non-anchor nodes autonomously derive their locations based on information from the anchors and neighboring nodes via secure range-free localization techniques such as [3], [12], [15]. The LBKs are also generated on the nodes' own. To enable this, each node s_i is in addition preloaded with the third initial key K_I^3 whereby to generate its LBK $LK_{s_i} = H(K_I^3|ID_{s_i}|pos_{s_i})$, where pos_{s_i} refers to the binary presentation of node s_i's position. This ends the key establishment phase (bootstrapping phase), and the node permanently removes from its storage the initial keys K_I^1, K_I^2, K_I^3 and all the IBKs of its neighbors computed in the neighbor discovery phase. Similar to [21], [22], this approach takes advantage of the fact that sensor nodes deployed in security-sensitive environments are usually designed to withstand break-in attacks at least for a short time interval T_{min} when captured by adversaries, and each node takes some time less than T_{min} to finish localization and generation of its LBK. After that, the node can no longer derive any keys.

2) Range-based localization: In this approach, we assume that a group of mobile robots are dispatched to sweep across the whole ROI along pre-planned routes. Mobile robots have GPS capabilities as well as more powerful computation and communication capacities than ordinary nodes. Each robot is also equipped with the initial keys K_I^1 and K_I^3 and is designed to be tamper-resistant[1]. Since the localization time in this approach is hard to bound, and our LBK establishment scheme discussed below does not require sensor nodes deriving any keys, each sensor node can erase the initial keys K_I^1, K_I^2 and all the IBKs of its neighbors after the neighbor discovery phase. To localize a node, say s_i, a mobile robot MS_j first broadcasts a special *HelloLBK* message to announce its existence. If seeing such a message and still uninitialized, node s_i responds with its ID_{s_i} and uses its IBK to generate MIC:

$$\begin{array}{lcl} MS_j \rightarrow * & : & MS_j, HelloLBK \\ s_i \rightarrow MS_j & : & ID_{s_i}, \mathsf{MIC}(IK_{s_i}, MS_j|ID_{s_i}) \end{array}$$

Upon receipt of the message, robot MS_j first uses its preloaded key K_I^1 to obtain IK_{s_i}, and then regenerates the MIC. If the result matches with what s_i sent, MS_j runs the secure range-based localization protocol given in [2] to get the location of s_i, and generates $LK_{s_i} = H(K_I^3|ID_{s_i}|pos_{s_i})$. After that, MS_j encrypts pos_{s_i} and LK_{s_i} with IK_{s_i}, and unicasts the ciphertext to node s_i:

$$\begin{array}{l} MS_j \rightarrow s_i : MS_j, \{pos_{s_i}, LK_{s_i}\}_{IK_{s_i}}, \\ \qquad\qquad \mathsf{MIC}(IK_{s_i}, ID_{s_i}|pos_{s_i}|LK_{s_i}). \end{array}$$

Upon reciept of the message, node s_i first uses its preloaded IBK IK_{s_i} to decrypt pos_{s_i} and LK_{s_i}, then regenerates the MIC. If the result matches with what MS_j sent, s_i saves pos_{s_i} and LK_{s_i} for subsequent use.

[1]Mobile robots are much fewer than ordinary sensor nodes and hence we can spend more on them by enclosing them in high-quality tamper-proof hardware and putting them under super monitoring.

D. Node Addition

Sooner or later new sensor nodes may need to be deployed in the ROI to maintain necessary connected coverage. When a flock of sensor nodes are re-deployed, the neighborhood authentication and PK establishment procedure between new nodes is the same as described in Section IV-B, and the localization and LBK establishment for new nodes is also the same as old ones. The authentication procedure between new and old nodes can be proceeded as follows:

When the old node s_o receives the *HelloNeig* message from the new node s_n, i.e., $ID_{s_n} > ID_{max}$, it replies with an ACK message including the plaintext of its ID, the ciphertext of its position and broadcast key encrypted with its IBK and the MIC generated by its LBK.

$$\begin{aligned} s_n \rightarrow * \quad &: \quad ID_{s_n}, HelloNeig, \mathsf{MIC}(IK_{s_n}, ID_{s_n}); \\ s_o \rightarrow s_n \quad &: \quad ID_{s_o}, \{pos_{s_o}\}_{IK_{s_i}}, \mathsf{MIC}(LK_{s_o}, pos_{s_o}). \end{aligned}$$

Upon receipt of the message, node s_n first generates the IBK of s_o from its ID, and then obtains pos_{s_o}. If range-free localization techniques are used, s_n can get its position information before it erases the initial keys. Therefore, s_n can first check whether $\|s_n - s_o\| \leq r_c$. If it holds, node s_n calculates s_o's LBK, and uses it to regenerate the MIC. If the result matches with what s_o sent, node s_n accepts s_o as an authenticated neighbor and calculates the pairwise key as $PK_{s_o s_n} = H(LK_{s_o}|ID_{s_n})$ and s_o's BK. If range-based localization techniques are used, s_n cannot get its own position before it erases the initial keys. Therefore, node s_n has to calculate $PK_{s_o s_n}$ and check the MIC first. If the result matches with what s_o sent, node s_n temporarily stores the information about s_o. When pos_{s_n} is obtained from the mobile robot, node s_n can then finally authenticate s_o by checking whether $\|s_n - s_o\| \leq r_c$ holds. After that, node s_n can reply the authenticated old nodes as follows:

$$\begin{aligned} s_n \rightarrow s_o \quad &: \quad ID_{s_n}, \{pos_{s_n}, BK_{s_n}\}_{PK_{so sn}}, \\ &\quad\quad \mathsf{MIC}(PK_{s_o s_n}, pos_{s_n}|BK_{s_n}). \end{aligned}$$

Upon receiving the message, node s_o can also calculate $PK_{s_o s_n}$, and use it and MIC to authenticate s_n. After the deployment of new nodes, the base station will broadcasts the new value of ID_{max} using μTESLA [17]. Each node will update this value accordingly for the next time node addition. Note that the compromised node cannot cheat others by change its ID into a large number in the node addition phase, since it does not have the corresponding IDK and PKs.

E. Some Discussion on LBSK

A desirable feature of LBSK is resistance to node capture. Each sensor node s_i is bounded with its position by its LBK, and its PKs shared with its neighbors. Even if a node is compromised and its key material is revealed, an adversary should not be able to gain control of other parts of the network by using this material. Its clones also cannot access other area of the network since they cannot establish the PKs with their neighbors. Therefore the compromise of nodes only result in a breach of security that is *constrained* within a small, localized part of the network.

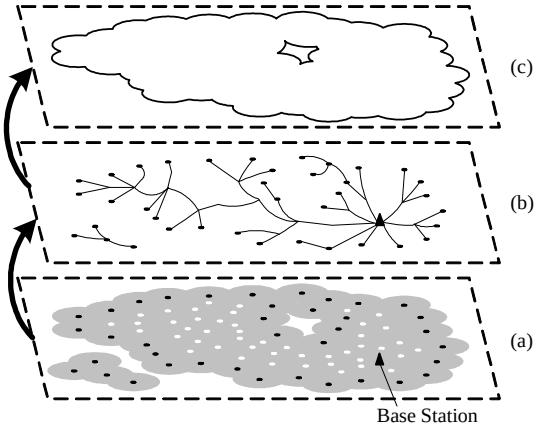

Fig. 1. **Basic operations of SCIP.** *(a) Each sensors takes BOND individually. (b) Boundary nodes (black dots in (a)) report themselves to the base station. (c) The BS reconstruct the coverage boundary. Note that the shaded area in (a) represents coverage of sensors. The shaded area at the left bottom corner in (a) is lost in (c) since it is not the connected coverage.*

Our LBSK scheme incurs the following computational overhead. For establishing a PK and a BK with each neighbor, one sensor node only needs to verify one MIC, compute one MIC evaluate two secure hash function to generate corresponding PK and BK. The computational overhead for LBK establishment is almost the same instead of having one more encryption operation. The communication overhead for establishing a PK only includes two messages, which have two fields: a node ID and a MIC. Both these messages can be easily fit into one packet. The communication overhead for establishing each LBK is within the same order for establishing PK since it only involves one-hop message exchanges. Our LBSK scheme establish three kinds of keys for each sensor node to keep. If a node has k neighbors, it needs to store two individual keys (IBK and LBK), k PKs, and $k+1$ BKs. The required space for storing preloaded keys is even smaller. Thus, the computational, communication, and storage overhead of our LBSK scheme is very small.

V. Secure Coverage Inference Protocol

In this section, we show how to integrate the BOND and the LBSK to provide the SCIP.

Fig. 1 shows the basic operations of our SCIP. In the first step, each node utilize BOND to locally decide whether it

is a boundary node based on the information gathered in the neighborhood monitoring phase. Secondly, those boundary nodes send their position information to the BS. If the BS can correctly receives these reports, it then has enough information to reconstruct the "image of the coverage".

Our design philosophy is that, since the minimum information required to describe the coverage is the positions of boundary nodes, we just need to detect boundary nodes. In other words, our scheme can ensure that, for the BS to reconstruct the "coverage image" without any distortion, the information transmitted from sensors to the BS is minimized. Also note that our BOND only involves local message exchanges. In a large-scale WSN, the overhead from local broadcasts is very small as compared to that from the end-to-end communications from sensor nodes to the BS. Therefore, our approach can save the precious energy of sensor nodes.

There are three kinds of communications involved in our SCIP: local broadcasts for neighborhood monitoring, boundary node self-reporting and returning ACKs from the BS to boundary nodes. In the following, we will explain how to secure them by exploiting the keys established in previous section.

A. Secure Neighborhood Monitoring and BOND

After the related keys are established, each node should broadcast its positions as soon as possible.

$$s_i \rightarrow * \quad : \quad ID_{s_i}, \{pos(s_i)\}_{BK_{s_i}}, \mathsf{MIC}(BK_{s_i}, ID_{s_i}|pos(s_i)),$$

Therefore, when a node is compromised, even if he wants to change his position in the one-hop range, it will be detected by its neighbors. In our SCIP, each node self-identifies whether it is a boundary node based on its trusted neighbor's information. One way to gather these information is to require each node periodically broadcasts its keep-alive message with the MIC generated by its broadcast key:

$$\begin{aligned} s_i \rightarrow * \quad : \quad & ID_{s_i}, \{Count(s_i)|alive\}_{BK_{s_i}}, \\ & \mathsf{MIC}(BK_{s_i}, ID_{s_i}|Count(s_i)|alive), \end{aligned}$$

where $Count(s_i)$ represents the number of messages it has already sent. In this approach, the communication overhead is small because a node only adds one MIC to each packet, and it exploits the broadcast advantage of the WSN. This periodic broadcasts can also be piggybacked with other data message node s_i needs to transmit. However, although this approach defends against outsider attacks in which the adversary does not hold any keys, insider attacks are possible after the adversary compromises a sensor node. The adversary could easily impersonate its neighbors with their broadcast keys since a broadcast key BK_{S_i} is shared between a node and all its neighbors. Note that the compromise of a sensor node only allows the adversary to launch impersonation attack in a two-hop zone of the compromised node s_i, because node s_i only has the broadcast keys of its one-hop direct neighbors. However, in some scenarios, it is sufficient for some compromised nodes to impersonate multiple dead nodes and to cheat the truly boundary nodes to believe themselves as interior nodes, so the neighborhood monitoring purely based on BKs is pretty insecure.

To further deter this attack, we present below a probabilistic challenge scheme. Each node challenges the authenticity of a received keep-alive message with a certain probability. More specifically, when node s_i receives a packet with broadcast key BK_{s_j} from (claimed) node s_j, it challenges node s_j for the authenticity of packet with probability p_c, using their pairwise key $PK_{s_is_j}$. The process is as follows.

$$\begin{aligned} s_i \rightarrow s_j \quad : \quad & ID_{s_i}, Noun_{s_i}, \mathsf{MIC}(PK_{s_is_j}, Noun_{s_i}); \\ s_j \rightarrow s_i \quad : \quad & ID_{s_j}, Noun_{s_j}, \mathsf{MIC}(PK_{s_is_j}, Noun_{s_j}|Noun_{s_i}). \end{aligned}$$

This scheme will be efficient if and only if the number of neighbors the node needs to challenge is small. For our BOND scheme, given that the nodes follows SPPP, each node only needs to check 4-6 neighbors in average, which is very suitable for this probabilistic challenge scheme.

B. Secure Boundary Node Self-Reporting

In our LBSK, every node s_i has two unique keys LK_{s_i} and IK_{s_i} that it shares pairwise with the BS. When node s_i identifies itself as boundary node, it will automatically generate a report $RP(s_i) = BN|Count(s_i)$ where $Count(s_i)$ denotes the number of packets s_i has already sent to the BS. Node s_i sends the following message to the BS:

$$\begin{aligned} s_i \rightarrow\rightarrow BS : ID_{s_i}, & \{pos_{s_i}|RP(s_i)\}_{IK_{s_i}}, \\ & \mathsf{MIC}(LK_{s_i}, ID_{s_i}|pos_{s_i}|RP(s_i)). \end{aligned}$$

Upon receipt of the report, the BS first regenerates IK_{s_i} based on node ID, decrypts the report and the position of the boundary node. Then the BS can obtain LK_{s_i} and regenerate the MIC. The BS can authenticate this report by checking whether the MIC matches what it received, since only the node in that specific position can generate the right MIC. Note that the node cannot fabricate reports from other positions since he does not have the related LBKs and intermediate nodes cannot get any information about pos_{s_i} or $RP(s_i)$ since in general s_j do not know the mapping between node ID and its position, and those information is end-to-end encrypted by IKs_i.

C. Secure Returning ACKs from the BS

Since the packet loss ratio is pretty high in the WSN and some compromised intermediate nodes may intentionally drop packets, boundary nodes need have some mechanisms to ensure that their reports have been received by the BS. Otherwise, they have to repeatedly resend their reports which causes energy waste. The native solution is to require the BS to return ACKs to boundary nodes individually and encrypt each ACK with corresponding IDK or LBK. Here, we design an energy-efficient approach for the BS to broadcast only one ACK to acknowledge multiple boundary nodes which also ensures that the adversary cannot reveal the acknowledged nodes from the ACK.

Let $s_1, s_2, \cdots, s_a$ be the a boundary nodes which the BS wants to acknowledge explicitly. Let $h_1, h_2, \cdots, h_b$ be the b hash functions of the Bloom filter, each with range $\{1, 2, \cdots, l\}$. Let $ack(t) = (b_1, b_2, \cdots, b_l)$ be a bit vector of length l. $ack(t)$ is the t-th Bloom filter used to indicate boundary nodes the BS wants to acknowledge, and it is initially set to $(0, 0, \cdots, 0)$. Then the BS goes through the following constructions: First, for every boundary node s_j, $1 \leq j \leq a$, the BS computes $\mathsf{MIC}(LK_{s_j}, t-1)$ where the $t-1$ denotes the number of Bloom filters the BS has already sent to the boundary nodes. Then, for $\forall i$, $1 \leq i \leq b$ and $\forall j$, $1 \leq j \leq a$, the BS computes $h_{ij} = h_i(\mathsf{MIC}(LK_{s_j}, t-1))$, and sets $b_{h_{ij}} = 1$. The BS can use a broadcast authentication protocol such as μTESLA [17] to authenticate the Bloom filter $ack(t)$ and the value t in the ACK.

When a boundary node s_j receives the $ack(t)$, it performs a membership test: it computes $\mathsf{MIC}(LK_{s_j}, t-1)$, hence $h_i(\mathsf{MIC}(LK_{s_j}, t-1))$ for $\forall i$, $1 \leq i \leq b$; if all of these positions are 1 in the $ack(t)$, then boundary node s_j knows its report has been received and acknowledged by the BS.

The adversary cannot get any information from the $ack(t)$ since if she does not have one node's LKB, she cannot do the membership test for any other node.

VI. Conclusions

In this paper, we propose a coverage inference protocol which can provide the BS an accurate and in-time measurement of the current connected coverage based on a novel computational geometric technique called LVPs. We also design the LBSK to provide location-aware security mechanisms to protect our basic measurement scheme. It has been shown that our SCIP is resilient to attacks from external attackers as well as to attacks from a subset of compromised nodes.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks (Elsevier) Journal*, 38(4):169–181, 393-422 2002.

[2] S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *INFOCOM'05*, Miami, FL, March 2005.

[3] W. Du, L. Fang, and P. Ning. LAD: Localization anomaly detection for wireless sensor networks. In *IPDPS'05*, pages 99–106, Denver, CO, Apr. 2005.

[4] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. In *INFOCOM'04*, Hong Kong, China, March 2004.

[5] S. Ganeriwal and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. In *Proc. of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Washington, DC, October 2004.

[6] A. Ghosh. Estimating coverage holes and enhancing coverage in mixed sensor networks. In *LCN'04*, Washington, DC, Nov. 2004.

[7] C. Hsin and M. Liu. Self-monitoring of wireless sensor networks. *Journal of Computer Communications special issue on Sensor Networks*, 29(4):462–476, February 2006.

[8] C. F. Huang and Y. C. Tseng. The coverage problem in a wireless sensor network. In *Proc. of the 2nd ACM international Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, San Diego, CA, September 2003.

[9] S. Kapoor and X. Li. Proximity structures for geometric graphs. In *WADS 2003*, Ottawa, Canada, July 2003.

[10] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley Sons Inc., New York, 2005.

[11] C. Karlof and D. Wagner. Secure routing in sensor networks: Attacks and countermeasures. *Ad Hoc Networks*, 1(2-3):293–315, September 2003.

[12] L. Lazos and R. Poovendran. Serloc: Secure range-independent localization for wireless sensor networks. In *ACM WiSe'04*, pages 21–30, Philadelphia, PA, Oct. 2004.

[13] M. Leonov. Polyboolean library (2004). http://www.complex-a5.ru/polyboolean/.

[14] X. Li, G. Calinescu, P. Wan, and Y. Wang. Localized delaunay triangulation with applications in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, Oct. 2003.

[15] D. Liu, P. Ning, and W. Du. Attack-resistant location estimation in sensor networks. Los Angeles, CA, Apr. 2005.

[16] A. Okabe. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, New York, 2000.

[17] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. Spins: Security protocols for sensor networks. In *ACM MobiCom 2001*, Rome, Italy, July 2001.

[18] G. Tsudik. Message authentication with one-way hash functions. *ACM SIGCOMM Computer Communication Review*, 22(5):29–38, October 1992.

[19] C. Zhang, Y. Zhang, , and Y. Fang. Localized coverage boundary detection for wireless sensor networks. In *QShine'06*, Waterloo, Canada, August 2006.

[20] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Network*, 1(1-2):89–123, January 2005.

[21] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 24(2):247–260, February 2006.

[22] S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *Proc. of the 10th ACM Conference on Computer and Communications Security (CCS 03)*, Washington, DC, October 2003.