Base Station Location Protection in Wireless Sensor Networks: Attacks and Defense

Juan Chen¹, Hongli Zhang¹, Xiaojiang Du², Binxing Fang¹, Yan Liu³, Haining Yu¹

¹Research Center of Computer Network and Information Security Technology, Harbin institute of technology, Harbin, China, e-mail: janechen.hit@gmail.com
²Dept. of Computer and Information Sciences, Temple University, Philadelphia, PA, USA, e-mail: dux@temple.edu
³School of Astronautics, Harbin Institute of Technology, Harbin, 150001, China, e-mail: yan.liu_hit@yahoo.com

Abstract—A base station (BS) is the controller and the data receiving center of a wireless sensor network. Hence, a reliable and secure BS is critical to the network. Once an attacker locates the BS, he can do a lot of damages to the network. In this paper, we study the BS location protection issue. First, we present a new attack on BS: the Parent-based Attack Scheme (PAS). The PAS can locate a BS within one radio (wireless transmission) range of sensors. Different from existing methods, the PAS determines the BS location based on parent-child relationship of sensor nodes. The PAS cannot be defended by existing BS protection schemes. To defend against the PAS, we design a new parent-free (PF) secure routing protocol for sensor networks. Our simulation results show that the PF protocol has small communication and computation costs, while ensuring the security of the BS.

Keywords – wireless sensor networks; base station; location protection

I. INTRODUCTION

As an important part of the Internet of Things, wireless sensor networks (WSNs) are becoming increasingly popular with applications ranging from habitat monitoring to battle field. In sensor networks, sensor placement is often driven by the need to sense certain phenomena. Low-density sensor networks are suitable in circumstances with easy node replacement, while applications such as structural health monitoring require high density deployments [1]. A sensor network with 40 or more neighbors per node is generally considered as a high-density sensor network [2].

As the controller and also the data receiving center, a base station (BS) is critical to the entire network. It could cause severe damages to the network if an adversary is able to locate the BS. Existing BS location attacks include packet-tracing attack [3], rate monitoring attack [3] and Zeroing-In attack [4]. Packet-tracing attack and rate monitoring attack can be defended by fake message injection or multi-path routing. Zeroing-In attack cannot be launched to routing protocols that do not use hop count information.

In this paper, first we present a new attack on BS location, called the Parent-based Attack Scheme (PAS). The PAS determines BS location by using parent-child information of sensor nodes. Our theoretical analysis and simulation results show that the PAS can locate BS within one sensor radio range, which is sufficient to find the BS. The existing BS protection schemes cannot defend the PAS. To protect BS from the PAS, we design a new parent-free (PF) secure routing protocol for sensor networks. PF successfully camouflages the parent information of each sensor node. Our performance analysis shows that PF can defend the PAS, and has small communication and computation costs. Furthermore, PF can defend against the Zeroing-In attack [4] because under PF nodes do not have hop-count information. PF can also be combined with some existing BS location protection schemes [7, 9] to defend against the packet-tracing [3] and rate monitoring attacks [3].

The rest of the paper is organized as follows. We give the network and attack model in Section II. We discuss the PAS in Section III, and show the effectiveness of the PAS in Section IV. We present the PF secure routing protocol in Section V, and evaluate its performance in Section VI. Finally, we draw our conclusion in Section VII.

II. THE NETWORK AND ATTACK MODEL

Our network model is the same as that in existing BS location protection routing protocols (e.g., [5, 6]). The entire network consists of one BS and a large number of sensor nodes. Without loss of generality, we assume that sensor nodes are distributed uniformly throughout the network. BS can be placed anywhere. A sensor has limited computation, power, and storage resources. BS is not constrained in power, communication and computation capabilities. We do not assume a specific MAC protocol. Each sensor node has a transmission range $R$. If the distance between two sensor nodes is no more than $R$, the two nodes are neighbors and they can communicate with each other directly. Each node has a parent set and transmits its message to one of its parent with a certain probability.

Next, we discuss the attack model. There may be multiple colluding adversaries in the network. An adversary may have more powerful hardware than a sensor. Specifically, an adversary may have the following capabilities:

- **Eavesdropping** - An adversary is able to receive messages sent by sensors within his monitoring range.
• **Active attacks** - An adversary can capture a sensor, compromise it and then obtain all information stored in the sensor.

• **Node localization** - An adversary is able to estimate the location of a node, by using existing localization schemes, such as the angle of arrival and/or the signal strength [12].

• **Colluding** - Several adversaries may collude with each other to infer the BS location.

### III. THE PARENT-BASED ATTACK SCHEME

#### A. Overview of the PAS

The PAS determines the location of a BS by parent sets of some nodes. Let \( R_{\text{opt}}(n_i) \) be the line passing through node \( n_i \) and BS. For any two nodes, say \( n_i \) and \( n_j \), if \( R_{\text{opt}}(n_i) \) and \( R_{\text{opt}}(n_j) \) intersect, then the intersection is the location of the BS. Hence, by obtaining \( R_{\text{opt}}(n_i) \) and \( R_{\text{opt}}(n_j) \), an adversary can locate BS. An adversary may find several locations close to \( R_{\text{opt}}(n_i) \) and generate a fitted line that approximates \( R_{\text{opt}}(n_i) \). More general, if there are \( m \) \((m \geq 2)\) adversaries, they can generate \( m \) fitted lines, compute the intersections and then estimate the location of BS from these intersections. Specifically, the PAS consists of three steps:

1) Location sampling. The \( i \)-th \((1 \leq i \leq m)\) adversary, say \( A_i \in \bar{A} \), stays at a location close to node \( n_i \). \( A_i \) tries to find \( h \) \((h \geq 1)\) locations around \( R_{\text{opt}}(n_i) \) via passive eavesdropping or active attacks (e.g., compromising the node) on some nodes.

2) Line fitting. \( A_i \) performs a least-square linear regression and generates a best fit line for \( h+1 \) locations including the location of \( n_i \) and the \( h \) sampled locations obtained by step 1).

3) BS location estimation. The \( m \) adversaries place themselves at different spots. They each perform step 1) and 2). After that, they generate \( m \) fitted lines and calculate the estimated location of BS—referred to as the EBSL (Estimated Base Station Location).

#### B. Location Sampling

The location sampling process is to find \( h \) locations close to \( R_{\text{opt}}(n_i) \). Denote \( U \) as a set of node locations, and denote \((x_j, y_j)\) as the \( j \)-th element (location) in \( U \). Denote \( P_i \) as the set of \( n_i \)'s parent nodes. First, we present a few definitions, Lemmas, and Theorems.

**Definition 1**: Let \( CM(U)=(x, y) \), where \( x \) and \( y \) are computed by Equation 1 and 2, respectively.

\[
x = (1/|U|) \sum_{j=1}^{|U|} x_j \quad (1)
\]

\[
y = (1/|U|) \sum_{j=1}^{|U|} y_j \quad (2)
\]

**Definition 2**: Node\((f)\) is a node placed at location \( f \).

**Definition 3**: NodeSet\((U)\) is a node set where each node is placed at a distinct location in \( U \), and \( U \) is the location set.

**Definition 4**: Define \( f_{\text{loc}}(n_i, h) \) as the \( h \)-th \((h \leq h_i)\) order critical location of node \( n_i \), where \( h_i \) denotes the shortest hop count between \( n_i \) and BS. Denote \( L_{\text{parent}}^{(i)} \) as the set of locations of \( n_i \)'s parent nodes.

1) If \( h=1 \), \( f_{\text{loc}}(n_i, h) \) is the location in \( L_{\text{parent}}^{(i)} \) which is closest to \( CM(L_{\text{parent}}^{(i)}) \).

2) If \( h \geq 2 \), \( f_{\text{loc}}(n_i, h) \) is the first order critical location of Node\((f_{\text{loc}}(n_i, h-1))\).  

**Definition 5**: Let \( f_{\text{cm}}(n_i, h) \) be the \( h \)-th order barycenter (center of mass) location of node \( n_i \).

1) If \( h=1 \), \( f_{\text{cm}}(n_i, h) \) is \( CM(L_{\text{parent}}^{(i)}) \).

2) If \( h \geq 2 \), \( f_{\text{cm}}(n_i, h) \) is the first order barycenter location of node Node\((f_{\text{loc}}(n_i, h-1))\).

**Definition 6**: Define set \( F_{\text{cm}}(n_i, h) = \{ f_{\text{cm}}(n_i, j) | 1 \leq j \leq h \} \).

**Definition 7**: Define set \( F_{\text{loc}}(n_i, h) = \{ f_{\text{loc}}(n_i, j) | 1 \leq j \leq h \} \).

![Fig.1: The area of \( n_i \)’s parent nodes](image)

**Theorem 1**: In a sensor network where nodes are uniformly distributed, \( f_{\text{cm}}(n_i, 1) \) is close to \( R_{\text{opt}}(n_i) \); as the node density increases, \( f_{\text{cm}}(n_i, 1) \) becomes closer to \( R_{\text{opt}}(n_i) \).

**Proof**: As shown in Fig.1, several circles with different radiiuses, say \( R, 2R, 3R, \ldots \), are centered at the BS. The \( q \)-th annulus is the area between the \((q-1)\)-th and \( q \)-th circles. We have that nodes in the \( q \)-th annulus are \( q \) hops away from BS, where \( q=2, 3, 4, \ldots \). Let node \( n_i \) be in the \((q+1)\)-th annulus. Thus, \( P_i \) is in the \( q \)-th annulus and are within the transmission range of \( n_i \). \( P_i \) is in the dotted area in Fig. 1. Since nodes are placed uniformly in the entire network, \( n_i \)'s parents are also uniformly distributed on both sides of \( R_{\text{opt}}(n_i) \). By definition 5, we have that the \( y \)-coordinate of \( f_{\text{cm}}(n_i, 1) \) is \( y = (1/w) \sum_{j=1}^{w} y_j \), where \( w \) is the number of \( n_i \)'s parents and \( y_j \) is the \( y \)-coordinate of the \( j \)-th parent. As shown in Fig.1, we set up a Cartesian Coordinate Plane with origin at node \( n_i \) and the two axis lines are: \( R_{\text{opt}}(n_i) \) and a line perpendicular to \( R_{\text{opt}}(n_i) \). Let \( y \)-coordinates of nodes in the parent-area range from \(-b \) to \( b \). Then \( y_1, y_2, \ldots, y_w \) are independent random variables following the uniform distribution in \([-b, b]\). Hence, we have the expectation of \( y_j - E(y_j) = 0 \), for \( 1 \leq j \leq w \). According to the law of large numbers, for any \( \varepsilon > 0 \), we have:

\[
\lim_{w \to \infty} P\left( \left| \frac{1}{w} \sum_{j=1}^{w} y_j \right| < \varepsilon \right) = 1
\]
When \( w \) gets large, the average of \( y_i \) converges to the expected value \( 0 \) with probability \( 1 \). This means that \( f_{cm}(n_1) \) is close to the line \( R_{opt}(n) \). Furthermore, we have \( w \geq \rho \), where \( \rho \) denotes the node density. Hence, as the node density increases, \( w \) also increases, and \( f_{cm}(n_1) \) becomes closer to the line \( R_{opt}(n) \). \( \square \)

**Lemma 1**: In sensor networks with nodes uniformly distributed, locations in \( F_{cm}(n,h) \) are close to \( R_{opt}(n) \) and they become closer to \( R_{opt}(n) \) as \( \rho \) increases.

**Proof**: 

1) When \( h=1 \), according to Theorem 1, \( f_{cm}(n_1) \) is close to \( R_{opt}(n) \) and \( f_{cm}(n_1) \) becomes closer to \( R_{opt}(n) \) as \( \rho \) increases. Hence, Lemma 1 is true when \( h=1 \).

2) Assume when \( h=j \) (\( 1 \leq j \leq h-1 \)), where \( i_t \) denotes the shortest hop count between \( n_t \) and BS, Lemma 1 is true. We have: \( f_{cm}(n_1, j) \) is closer to \( R_{opt}(n) \) as \( \rho \) increases. By definition 4, we have that \( f_{cm}(n_1, j) \) is the location of the node which is Node \((f_{cm}(n_1, h-1))'s \) parent and is closest to \( f_{cm}(n_1, j) \). Hence \( f_{cm}(n_1, j) \) is closer to \( R_{opt}(n) \) as \( \rho \) increases. Let \( l \) be the line passing through \( f_{cm}(n_1, j) \) and BS. Then, \( l \) approximates \( R_{opt}(n) \) as \( \rho \) increases. Since \( f_{cm}(n_1, j+1) \) is the first order barycenter location of Node \((f_{cm}(n_1, j))', \) according to Theorem 1 we have \( f_{cm}(n_1, j+1) \) is close to \( l \) and \( f_{cm}(n_1, j+1) \) becomes closer to \( l \) with increasing \( \rho \). Thus, \( f_{cm}(n_1, j+1) \) gets closer to \( R_{opt}(n) \) as \( \rho \) increases. Hence, the locations in \( F_{cm}(n_1, j+1) \) becomes closer to \( R_{opt}(n) \) as \( \rho \) increases. Lemma 1 is true when \( h=j+1 \). \( \square \)

**Theorem 2**: By passively monitoring (and/or actively compromising) node \( n_t \), an adversary can find \( f_{cm}(n_1, 1) \) and \( f_{cm}(n_1, 1) \).

**Proof**: By passive monitoring node \( n_t \) for enough time, an adversary can capture messages from both \( n_t \) and its neighbors, and infer their relationships and find out \( P_t \). Then he can locate nodes in \( P_t \) by some existing localization techniques, such as the angle of arrival (AOA) technique in [7]. If routing protocol is combined with security schemes such as fake-message injection [6], it is infeasible for a passive adversary to find out \( P_t \) as he cannot distinguish real messages from fake ones. In that case, an adversary may launch active attacks on node \( n_t \) and then obtain its secret information including \( P_t \) and the keys. After that, he can locate \( n_t \)'s parents by AOA [7]. With locations of \( n_t \)'s parents, the adversary can obtain \( f_{cm}(n_1, 1) \) and \( f_{cm}(n_1, 1) \). \( \square \)

**Lemma 2**: By monitoring or compromising node \( n_t \) and NodeSet \((f_{cm}(n_1, h-1)) \), an adversary can find \( F_{cm}(n_t, h) \).

**Proof**: 

1) When \( h=1 \), according to Theorem 2, an adversary can find \( f_{cm}(n_1, 1) \) by monitoring or compromising node \( n_t \);

2) When \( h=2 \), by definition 4 and 5, we have that \( f_{cm}(n, h) \) is the first order barycenter location of Node \((f_{cm}(n_1, h-1)) \). Therefore, an adversary can find \( f_{cm}(n_1, h) \) by monitoring or compromising node Node \((f_{cm}(n_1, h-1)) \) by Theorem 2. \( \square \)

According to Lemma 1, we have that locations in \( F_{cm}(n_t, h) \) (\( 1 \leq h \leq h_t \)) are close to \( R_{opt}(n_t) \), where \( h_t \) denotes the hop count of node \( n_t \). The location sampling process is completed if an adversary obtains \( F_{cm}(n_t, h) \). By Lemma 2, we have that an adversary can find \( F_{cm}(n_t, h) \) by monitoring or compromising \( n_t \) and NodeSet \((f_{cm}(n_1, h-1)) \).

**C. Line Fitting**

By the location sampling process above, the adversary \( A_i \) obtains \( U_i \) that includes \( h \) sampled locations and the location of \( n_t \). After that, \( A_i \) performs a least-square linear regression and generates a best fit line, say \( l_i = ax_i + b \), for locations in \( U_i \), where \( a \) and \( b \) are computed by (4) and (5), respectively. \( (x_i, y_i) \) denotes the \( j \)-th element in \( U_i \). By Lemma 1, locations in \( U_i \) are close to \( R_{opt}(n_t) \), hence \( l_i \) is close to \( R_{opt}(n_t) \).

\[
a = \frac{(\sum_{j=1}^{k+1} x_i y_j - k \sum_{j=1}^{k+1} y_j x_j) - (\sum_{j=1}^{k+1} x_j)^2}{k \sum_{j=1}^{k+1} x_j^2 - (\sum_{j=1}^{k+1} x_j)^2}
\]

\[
b = \frac{(\sum_{j=1}^{k+1} x_i y_j x_j^2 - k \sum_{j=1}^{k+1} y_j x_j x_i - \sum_{j=1}^{k+1} x_j y_j^2 + \sum_{j=1}^{k+1} x_j^2 y_j)}{k \sum_{j=1}^{k+1} x_j^2 - (\sum_{j=1}^{k+1} x_j)^2}
\]

**D. Estimation of BS Location**

If there are \( m \) adversaries and each of them performs the location sampling and line fitting process, then they can obtain \( m \) lines: \( L_i = \{l_i | 1 \leq i \leq m \} \). Let an estimation point be the intersection of two lines in \( L \). Suppose we have \( k \) (\( k \leq c_m^2 \)) estimation points from \( L \), where \( c_m^2 \) denotes the number of 2-combinations of \( m \) elements. It is possible that some estimation points (called noise points) are far away from the BS. There are two reasons for having noise points: (1) If the node density \( \rho \) is very low, for an adversary \( A_i \), one or two of his sampled locations might be away from \( R_{opt}(n_t) \) and \( l_i \) is also away from the BS, which causes some intersections of \( l_i \) are far away from the BS. (2) Two or more lines in \( L \) are nearly parallel. E.g., if \( R_{opt}(n_t) \) and \( R_{opt} \) (\( i \neq j \)) are nearly parallel to each other, then \( l_i \) and \( l_j \) are nearly parallel, and they will have no intersections or their intersections are far away from the BS. Let \( S \) be the set of the \( k \) estimation points. The PAS can reduce the number of noise points in \( S \) by clustering and then obtain a more accurate location of the BS [11]. The de-noising process is as follows:

1) Applying hierarchical clustering [11] on \( S \) and generate \( k' \) clusters with a given threshold;

2) Finding the maximum cluster, say \( c_{max} \), which includes the largest number of estimation points;

3) The estimated BS location is \( CM(c_{max}) \).

**IV. THE EFFECTIVENESS OF THE PAS**

We use the mean error \( \Delta d \) and the mean square error \( \Delta \delta \) to evaluate the performance of the PAS. \( \Delta d \) and \( \Delta \delta \) are computed by equations (6) and (7), and they are used to measure the attack accuracy. In (6) and (7), \( e \) is the number of attacks and \( d_i \) denotes the difference between estimated BS location and the actual BS location during the \( i \)-th attack. \( \Delta d \) and \( \Delta \delta \) are divided by the communication range \( R \) as in most existing localization works (e.g., [3, 8]).

\[
\Delta d = \frac{\sum_{i=1}^{e} |d_i|}{e * R}
\]

\[
\Delta \delta = (1/R) \times \sqrt{\sum_{i=1}^{e} (d_i - \Delta d)^2}/e
\]
The effectiveness of the PAS is validated by an event-driven sensor network simulator written in C++. For uniform sensor deployment, we divide the monitored area into small grids and place one node in a grid. To be more realistic, each node is not placed exactly in the center of a grid. For example, if \((x, y)\) is the center of a grid, a sensor node is placed at \((x+\varepsilon, y+\varepsilon')\), where \(\varepsilon\) and \(\varepsilon'\) are two uniform random variables on \((-0.5, 0.5)\). The BS is randomly placed in the network. The following results are averaged over 100 runs.

Our simulation uses a sensor network of 1024 nodes with \(h=1\) and the clustering threshold \(\eta\) is chosen as \(2.5R\). The mean error of the PAS is shown in Fig. 3, where the x-axis is the average number of neighbors of each node, and \(m\) is the number of adversaries in the network. Fig. 3 shows that as the number of adversary increases, the mean error decreases. Also, the mean error decreases when the number of neighbors increases. This is consistent with Lemma 1. When the average number of neighbors is over 40, adversaries can locate the BS with an accuracy of one radio range by passive monitoring or active compromising 8 nodes. However, the situation is different in low-density networks.

![Fig. 3. Mean error vs number of neighbors and adversaries](image)

Fig. 4 shows the mean error for varying the network size (number of sensors) with \(n=36, h=1, \eta=2.5R\) and \(m=12\), where \(n\) denotes the average number of neighbors. As the network size grows, we notice that the mean error increases in general. It is also observed that the mean error increases significantly when the network size is more than 1024. Fig. 5 shows the mean square error for varying number of neighbors with \(N=1024, h=1, \eta=2.5R\) and \(m=12\). It is observed that the larger the number of neighbors, the less the mean square error, which indicates that the PAS is more robust when the number of neighbors is large.

![Fig. 5. Mean square error vs number of neighbors.](image)

![Fig. 6. Mean error vs clustering threshold.](image)

Fig. 6 shows the mean error for varying \(\eta\) and \(h\). In this simulation, the parameters are set as follows: \(N=1024, n=36\) and \(g=12\), where \(g\) denotes the total number of nodes been attacked and \(g=\eta h\). Fig. 6 shows that \(\Delta d\) decreases when \(h\) becomes smaller, which indicates that given a fixed total number of nodes been attacked (i.e., given \(g\)), the attack accuracy is high even if each adversary only attacks a small number of nodes. Also, the results show that \(\Delta d\) has the lowest value when \(\eta=2.5R\). Note that \(\eta=0\) means the PAS without clustering.

To sum up, the above simulation results show that the PAS can locate the BS with high accuracy (e.g., within one-radio range) by attacking only a small number of nodes (e.g., 8 nodes).

V. THE PARENT FREE ROUTING PROTOCOL

As the PAS is based on parents’ locations, it will be infeasible for an attacker to find out the BS location if no sensor stores its parents’ information. Based on the above principle, we propose a parent free (PF) routing protocol to defend the PAS attack. The main idea of PF is as follows: Each node, say \(n_i\), has \(u\) onion packets each of which denotes a route from \(n_i\) to BS. Node \(n_i\) sends messages to BS by onion packets. As node \(n_i\) has no information about its parents, an adversary cannot find out \(n_i\)’s parents by compromising \(n_i\). Furthermore, in PF, two successive nodes in a route may not be parent-child, i.e., the next forwarding node may not be the parent of the previous one in a route. Therefore, even an adversary find out that a message has been transmitted from one node to another, he is not sure whether the latter is the
parent of the former. Hence, PF can defend the PAS attack. PF consists of two phases: network initialization and message sending. We present the details of PF below.

A. Network Initialization

Assume the network is secure (e.g., no attacks) for a short time period after sensor nodes are deployed. This is a common assumption used by several literatures (e.g., [12]). During this period, the communications among sensor nodes are secure. Before deployment, each node \( n_i \) is preloaded with several parameters: node ID - \( i \), keys \( k_i \) and \( k_{\text{BS}} \). \( k_i \) is \( n_i \)'s broadcast key which is shared between \( n_i \) and its neighbors. And \( k_{\text{BS}} \) is shared between \( n_i \) and BS. After deployment, BS generates and then sends \( u \) onion packets to \( n_i \) by the following two steps:

1) Topology discovery. BS first sends out a broadcast message to all nodes in the network. When each node receives the broadcast message, it updates the hop count and also includes the following in the message: its broadcast key, parent set \( P_i \) and non-parent set \( \bar{P}_i \) (\( \bar{P}_i = N_{\text{net}} - P_i \) ...), where \( N_{\text{net}} \) denotes the neighboring nodes of \( n_i \). After the broadcast, each node (say \( n_i \)) obtains the above information from its neighbors. Then, \( n_i \) sends \( P_i \) and \( \bar{P}_i \) to BS. Thereafter, each node deletes \( P_i \) and \( \bar{P}_i \).

2) Onion packets generation. For each node, say \( n_i \), BS generates \( u \) onion packets \( R_{i} = \{ r_{i}^{(1)}, r_{i}^{(2)}, \ldots, r_{i}^{(u)} \} \) and sends \( R_i \) to \( n_i \). For a route: \( a \rightarrow b \rightarrow \cdots \rightarrow n_i \rightarrow \text{BS} \), \( r_{i}^{(v)} \) has the form: \( E_{k_{\text{BS}}} (a \| E_{k_{\text{BS}}} (b \| \cdots ) \| P_{A}) \). Specifically, \( r_{i}^{(v)} (1 \leq v \leq u) \) is computed as follows:

- **Route Discovery.** First, \( n_i \) is chosen as the current node. Then, BS selects the first node in route \( r_{i}^{(v)} \), say \( n_j \), from \( P_i \) and \( \bar{P}_i \) with probability \( p \) and \( 1-p \) respectively. Next, \( n_j \) is chosen as the current node and BS repeats the above node selection process. The node selection process is repeated until BS is reached.

- **Duplicate Route Deletion.** If \( r_{i}^{(v)} \) is the same as some previously discovered route, BS runs the route discovery process again and tries to find a new route.

- **\( r_{i}^{(v)} \) Generation.** \( r_{i}^{(v)} \) is an onion packet with multi-layer encryptions. For example, if \( r_{i}^{(v)} \) goes through node \( n_a \) and \( n_b \) to reach BS, then \( r_{i}^{(v)} \) has the form \( E_{k_{\text{BS}}} (a \| E_{k_{\text{BS}}} (b) \| P_{A}) \), where \( P_{A} \) is a padding, which makes all onion packets of \( n_i \) have the same size.

B. Message Relay

Suppose node \( n_i \) is a source node and wants to send a message \( M_i \) to BS. \( n_i \) chooses an onion packet \( r_{i}^{(v)} \) randomly from \( R_i \) and broadcasts \( M_i \) with the form \( i \| E_{k_i} (r_{i}^{(v)} \| E_{k_{\text{BS}}} (data)) \). For \( \forall n_j \in N_{\text{net}} \), if \( n_j \) receive \( M_i \), \( n_j \) decrypts \( M_i \) and gets \( r_{i}^{(v)} \). Next, \( n_j \) tries to decrypt \( r_{i}^{(v)} \) by \( k_{n_j} \). If \( n_j \) cannot decrypt \( r_{i}^{(v)} \) successfully, \( n_j \) discards \( M_i \). Otherwise, \( n_j \) transmits the message to its neighbors with the form \( M_j = j \| E_{k_j} ((r_{i}^{(v)})' \| E_{k_{\text{BS}}} (data)) \), where \( (r_{i}^{(v)})' \) has the same length as \( r_{i}^{(v)} \). \( (r_{i}^{(v)})' \) is firstly decrypted from \( r_{i}^{(v)} \) and then padded by random bits. For example, if \( n_j \) receives an onion packet \( E_{k_{\text{BS}}} (s \| E_{k_{\text{BS}}} (s \| E(...) \| P_{A}) \) from \( n_i \), \( n_j \) decrypts the packet and obtains \( E_{k_{\text{BS}}} (s \| E(...) \) then \( n_j \) adds a new padding - \( P_{A}' \).

VI. PERFORMANCE EVALUATION

In this Section, we evaluate the performance of our PF routing protocol, including the communication cost, computation cost, and security.

A. Communication Cost

The communication cost is the total number of transmissions of a process. The communication cost of PF includes the message transmissions during the network initialization phase and the message sending phase. Note that we do not include the communication cost of the initial broadcasting since it is the same as other existing routing protocols (e.g., [9, 10]). After the broadcast, each node, say \( n_i \), sends \( P_i \) and \( \bar{P}_i \) to BS through the shortest path routing. The communication cost for this is:

\[
Q = \sum_{q=1}^{h_{\text{max}}} (\bar{N}_q q)
= n \sum_{q=1}^{h_{\text{max}}} (2q - 1)q
= 2n \sum_{q=1}^{h_{\text{max}}} q^2 - n \sum_{q=1}^{h_{\text{max}}} q
= n h_{\text{max}} (h_{\text{max}} + 1)(4h_{\text{max}}^2 - 1)/6
\]

where \( \bar{N}_q \) is the number of nodes with hop count \( q \), \( n \) denotes the average number of neighbors and \( h_{\text{max}} \) denotes the max hop count. If \( N \) nodes are uniformly distributed in the network, we have \( h_{\text{max}} = \sqrt{N/n} \) and \( Q = n \frac{n}{N} \left( \frac{n}{N} + 1 \right) \left( 4 \frac{n}{N} - 1 \right) / 6 \).

Thereafter, BS sends \( u \) onion packets to each node and the communication cost is also \( Q \). In all, we have the total communication cost \( 2Q \) in the initialization phase.

In the message sending phase, if a source node \( n_i \) sends a message to BS, the communication cost is \( h_i + 2h_i(1-p) \).

B. Computation Cost

The computation cost for PF is low since PF only uses symmetric encryption. The computation during the network initialization phase is a one-time operation and it is done by the base station where power and computational resource are abundant. During the message sending phase, two encryption operations are needed if a source wants to send a message to BS. In additional, whenever a node transmits a message, it needs three decryption/encryption operations with two for message verification and one for message transmission.

C. Security Analysis

PF is robust to the PAS attack as adversaries cannot find out parents of any node. An adversary could stay close to node \( n_i \), monitor and obtain messages exchanged between \( n_i \) and its neighbors. Also, the adversary could compromise \( n_i \) and obtain all its secret information. However, he still cannot find out \( P_i \), even though he is able to infer the transmission relationship between node \( n_i \) and its neighbors. This is because the next forwarding node of \( n_i \) may not be \( n_i \)'s parent (according to the
route discovery process). Furthermore, PF can defend against the Zeroing-In attack [4] because in PF nodes do not have hop-count information. It is also easy to combine PF with existing BS location protection schemes [7, 9] to defend against the packet-tracing [3] and rate monitoring attacks [3].

VII. CONCLUSION

In this paper, we studied the BS location protection problem from both the attack and defense sides. First, we presented a new BS attack scheme: the Parent-based Attack Scheme (PAS). Our theoretical analysis and experiments showed that the PAS can locate a BS within one sensor radio range. Existing BS protection schemes cannot defend the PAS. To protect a BS from the PAS, we designed a novel parent-free (PF) secure routing protocol for sensor networks. Our simulation results showed that the PF protocol can protect the BS location, and it has small communication and computation costs. Furthermore, PF can defend against several other attacks in sensor networks.

ACKNOWLEDGMENT

This research was supported in part by the China National Basic Research Program (973 Program) under grants 2011CB302605 and 2007CB311101, the China National High Technology Research and Development Program (863 Program) under grant 2010AA012504 and 2011AA010705; and by the US National Science Foundation under grants CNS-0963578, CNS-1002974, CNS-1022552, and CNS-1065444, as well as the US Army Research Office under grant W911NF-08-1-0334.

REFERENCES