An Efficient Secure DV-Hop Localization for Wireless Sensor Network

Xiaole Liu¹, Rui Yang² and Qingmin Cui¹

¹College of Computer, Henan Institute of Engineering, Zhengzhou 451191, China
²College of Science, Henan Institute of Engineering, Zhengzhou 451191, China
xzhn_2008@163.com

Abstract

Localization algorithm is an important and challenging topic in Wireless Sensor Networks (WSNs), especially for the applications requiring the accurate position of the sensed information. Various algorithms have been proposed to obtain the location of sensor nodes. However, most of existing location algorithms assumes a non-adversarial environment. The position estimation accuracy decreases drastically when some of the sensor nodes are compromised. In this paper, we develop a secure localization scheme to resist the attack on the DV-Hop scheme, to mitigate the impact of such attacks. In our scheme, the flooding packets in the DV-Hop will be authenticated and the weight of beacon nodes will be used to abate the effect of nodes capture attack. Analysis and simulation results demonstrate that the proposed can not only against nodes compromised attack effectively but also achieve comparable localization performance.

Keywords: Wireless sensor networks; Secure localization; DV-Hop algorithm

1. Introduction

Wireless Sensor Networks (WSNs) involve many different technologies such as communication, sensing and computing, which now becoming a very important research areas. The knowledge of localization of sensor nodes in WSN is a fundamental issue for many applications such as environment monitoring, emergency rescue and battlefield surveillance, etc [1]. Besides many fundamental techniques in WSNs, e.g. geographical routing [2], require the position of unknowns nodes.

Recently many algorithms have been designed to solve the nodes’ localization problem, which are categorized into two categories: range-based [3,4] and range-free[5,6]. Range-based algorithms rely on absolute distance from transmitting to receiving sensor nodes. In contrast, range-free algorithms do not need absolute range information or angle between sensor nodes, only need network connectivity and other information, so the range-free algorithms are more economical, cost-effective, and feasible for the large-scale wireless sensor networks. The accuracy is less than the range-based but satisfies many applications’ requirements.

When WSNs are deployed in real world, the environments may untrustworthy, and even may be hostile with presence of malicious adversaries, e.g. in the battlefield related applications. In these environments, they are vulnerable to various attacks during and after its deployment. The attacks, which can threaten the localization of the nodes in a hostile WSN, can generally be classified into two categories, external attacks and internal attacks[7,8]. External attacks can distort network behaviors without obtaining the system’s authorization, while internal attacks are authenticated ones and thus more devastating to the security of the
In a hostile environment, adversary can mount physical attacks on a sensor node after it is deployed and read secret information from its memory. Therefore, nodes compromised attack is an internal attack and can deteriorate the localization dramatically.

Among many rang-free localization algorithms, DV-Hop[4] is a neat scheme which worth further investigations. Its basic idea realize on transforming the distance to all beacon nodes from hops to meters by using computer average size of a hop. The advantages of the DV-Hop scheme are that it does not need any sophisticated hardware for the distance measurement and thus, it is free from range measurement errors. However, the DV-Hop technique introduces errors that propagated to the computation of a node’s location. To decrease the localization error in DV-Hop algorithm, many improved method was proposed in [11, 12]. But these improved methods only focused on the accuracy of localization and didn’t consider the security. As a result, if WSNs have been attacked, the sensor node location estimation will be far away from precision. In this paper we propose a secure DV-Hop algorithm based on the DV-Hop localization scheme.

The remainder of the paper is organized as follows. Section 2 reviews the related work on the secure localization. In Section 3, we describe, the DV-Hop algorithm and weighted least square method. In section 4, the proposed secure scheme was given in detail. Simulation and analyses are shown in section 5. Finally, the conclusions are given in section 6.

2. Related Work

Labraoui et al. [13] propose a Wormhole-free DV-hop Localization scheme (WFDV) to defend wormhole attack in reactive countermeasure. Simulations show that the proposed solution is effective in detecting and defending against wormhole attacks with a high detection rate.

Garg et al. [14] propose a secure localization algorithm which combines iterative gradient descent with selective pruning of inconsistent measurements to achieve high localization accuracy. Results show that the proposed algorithm utilizes fewer computational resources and achieves accuracy better than or comparable to that of existing schemes. The proposed secure localization algorithm can also be used in mobile sensor networks, where all nodes are moving, to estimate the relative locations of the nodes without relying on anchor nodes.

Han et al. [15] propose a novel scheme called two-step secure localization (TSSL) stand against many typical malicious attacks, e.g. wormhole attack and location spoofing attack. TSSL detects malicious nodes step by step. First, anchor nodes collaborate with each other to identify suspicious nodes by checking their coordinates, identities and time of sending information. Then, by using a modified mesh generation scheme, malicious nodes are isolated and the WSN is divided into areas with different trust grades. Finally, a novel localization algorithm based on the arrival time difference of localization information is adopted to calculate locations of unknown nodes.

He et al. [16] propose a reputation-based security scheme for sensor localization to improve the security and the accuracy of sensor localization in hostile or untrusted environments. In the proposed scheme, the reputation of each beacon node is evaluated based on a reputation evaluation model so that regular sensor nodes can get credible location information from highly reputable beacon nodes to accomplish localization.
3. Preliminaries

In this section, we will describe the DV-Hop localization approach, and the weight least square in brief.

3.1 The DV-Hop Algorithm

DV-Hop algorithm is a classical localization method for WSNs. The algorithm implementation consists of three stages.

In the first stage, each beacon node broadcasts a message to be flooded throughout the network that contains its location information with hop-count value initialized to 0. On receiving the beacon packet, each sensor node maintains the position of the beacon \((x_i, y_i)\) and the minimum hop-count \(hop_i\) to the particular beacon node \(i\). If a received message contains less hop-count value to a particular beacon node, the corresponding hop-count \(hop_i\) will be replaced with the information in this message. And this message is flooded outward with hop-count values incremented y one. On the contrary, if the received message contains a higher hop-count value to a particular beacon node, this message will be ignored.

In the second stage, once a beacon gets hop-count value to other beacons, it estimates an average size for one hop. The average hop-size can be computed by beacon \(i\) as follow:

\[
HopSize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} hop_j}
\]

(1)

where \((x_i,y_i)\), \((x_j,y_j)\) are coordinates of beacon \(i\) and beacon \(j\), \(hop_j\) is the hops between beacon \(i\) and beacon \(j\). Once the average hop-size is calculated, each beacon node broadcasts its hop-size to network using controlled flooding. After receiving hop-size, all nodes can derive the physical distance to the beacon by using Eq(2).

\[
d_{ij} = HopSize_j \times hop_j
\]

(2)

where \(hop_{ij}\) is the minimal hops between beacon \(i\) and unknown node \(j\). Based on this method, each beacon node can convert the distance to physical distance.

In the third stage, each unknown node computes its location coordinate. Let \((x, y)\) be the unknown node \(P\) location and \((x_i, y_i)\) the known location of the \(i\)'th beacon node receiver. Let \(d_i\) be the \(i\)'th beacon node distance to unknown nodes \(P\), then we have

\[
d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}
\]

(3)

After obtaining all the distance information, each sensor node can conducts the triangulation or maximum likelihood estimation [13] to estimate its own location.

3.2 Weighted Least Square Method

Suppose there are \(n\) beacon nodes \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\) and their weight value are \(w_1, w_2, \ldots, w_n\) respectively. There is a unknown nodes \(D\) \((x, y)\) and the distance between the unknown nodes \(D\) and the beacon nodes are \(d_1, d_2, \ldots, d_n\) respectively. Then the goal function can be defined as following:

\[
f(x, y) = \min \sum_{i=1}^{n} w_i \left( \sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i \right)^2
\]

(4)

where, \(w_i\) is the weight of beacon node \(i\).

So there exits the following formulas
\[
\begin{align*}
\left\{ w_i[(x - x_i)^2 + (y - y_i)^2] &= w_i d_i^2 \quad (E_i) \\
\right.
\end{align*}
\]
Do the following calculation in (5)
\[
E_i \times w_i^n - E_n \times w_i^2, \quad 1 \leq i < n
\]
Then we get
\[
\left\{ 2w_i^2 w_n^2(x_i - x_n)x + 2w_i^2 w_n^2(y_i - y_n)y = w_i^2 w_n^2(x_i^2 - x_n^2 + y_i^2 - y_n^2 + d_i^2 - d_n^2) \\
\right.
\]
Let
\[
X = \begin{bmatrix} x \\ y \end{bmatrix}
\]
\[
A = \begin{bmatrix} w_1^2 w_n^2(x_1 - x_n) & w_1^2 w_n^2(y_1 - y_n) \\
\vdots & \vdots \\
2w_{n-1}^2 w_n^2(x_{n-1} - x_n) & 2w_{n-1}^2 w_n^2(y_{n-1} - y_n) \\
\end{bmatrix}
\]
\[
b = \begin{bmatrix} w_1^2 w_n^2(x_1^2 - x_n^2 + y_i^2 - y_n^2 + d_i^2 - d_n^2) \\
\vdots \\
w_{n-1}^2 w_n^2(x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_{n-1}^2 - d_n^2) \\
\end{bmatrix}
\]
Then
\[
AX = b
\]
From formula (10), we can get the position of unknown nodes \(D\)
\[
X = (A^T A)^{-1} A^T b
\]

4. The Proposed Scheme

In our scheme, we assume that all beacon nodes will send their position to the base station secretly, and the base station will broadcast the position of all beacon nodes to the network using broadcast authentication scheme[17]. And we also assume there are some intrusion detection schemes in the sink node. The proposed scheme includes four phase.

1) The Initialization Phase

Before the sensor nodes deployed, the sink node generates random keys for each beacon sensor node and applies the hash function to generate hash chain. Suppose the random key for beacon node \(u\) is \(R_u\), the hash function is \(H\), then the key chain is \(R_{u}, R_{i}, \ldots, R_n\), here \(R_{i+1} = H'(R_i)\) and \(n\) is the maximum hop count between any beacon node and unknown node.

The base station will store the hash function and the last key \(R_n\) of the each key chain into all sensor nodes as the authentication key of the beacon node.
2) Hop-count Computation

The goal of this phase is that all sensor nodes get the minimal hop-count to each beacon node. In this phase all beacon nodes broadcast its information to its neighbor nodes. Let sensor node \( B \) is a beacon node. All sensor node \( i \) will broadcast a message to its neighbors

\[ S_i \rightarrow ^*: ID(B), P(B), h_i(B), R_i(B) \]

Here, \( ID(B) \) is the identity of the beacon node \( B \), \( P(B) \) is the position the beacon nodes \( B \), \( h_i(B) \) is the hop count between the sensor node \( i \) and the beacon node \( B \), \( R_i(B) \) is the authentication key of beacon node \( B \) in sensor node \( i \). If sensor node \( i \) is a beacon node \( h_i(B)=0 \) and \( R_i(B) \) is the authentication key of the beacon node.

When sensor node \( j \) receive the broadcast message from sensor node \( i \), there are two cases needed to be considered:

Case 1: \( h_i(B)<h_j(B)-1 \). In this case sensor node \( j \) will compute \( h_i(B)-h_j(B) \) and compare the authentication key of beacon node \( B \) in its memory. If this two value are equal, the sensor node \( j \) will change the hop count to beacon node \( B \) to \( h_i(B)+1 \), and the authentication key of beacon node \( B \) to \( R_i(B) \). At last the sensor node \( j \) will broadcast a message to its neighbors as above.

Case 2: \( h_i(B) \geq h_j(B)-1 \). In this case, the sensor node \( j \) will simple drop the receiving message from sensor node \( i \) and do nothing.

3) Hop-size and Weighted Computation

In this phase, all beacon nodes will compute the distance to the beacon nodes and the weight of beacon nodes. To achieve there goals, the beacon node \( i \) do the following computations.

Step 1: In this step, every sensor node estimates an average size for one hop and the hop-size weight of other beacon nodes as following.

\[ \text{1. Computing } HoS_j = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_{ij}} \]  \hspace{1cm} (12)

where \((x_i,y_i), (x_j,y_j)\) are coordinates of beacon node \( i \) and beacon node \( j \), \( h_{ij} \) is the hops between beacon \( i \) and beacon node \( j \).

Step 2: Computing \( HoS_i \), which is the median of \( HoS_{ij} \), \( 1 \leq j \leq N \), where \( N \) is the number of beacon nodes in the network.

Step 3: Computing the hop-size weight of beacon nodes \( j \) as follows:

\[ \text{wh}_{ij} = \frac{1 - (HoS_{ij} - HoS_i)^2}{r^2} \]  \hspace{1cm} (13)

Here, if \( wh_{ij} < 0 \) then \( wh_{ij}=0 \), where, \( r \) is the communication range of nodes.

4. Broadcasting the \( <HoS_i, wh_{ij}> \) to all nodes.

Step 2: After the step 1, all of sensor nodes get the \( HoS \) and \( wh \) of all beacon nodes. Sensor node \( i \) gets the \( HoS \) of beacon nodes \( j \) which is the median of \( HS_k \) and \( W_k \) of beacon nodes \( j \), which is the median of \( W_k \), \( 1 \leq k \leq N \).

Step 3: At this step, unknown sensor nodes \( i \) computes the distance to the beacon nodes \( j \)

\[ d_{ij} = h_{ij} \times HS_j \]  \hspace{1cm} (14)

4) Location Estimation

An unknown sensor node can calculate its location when it has get estimate distance to at least three beacons and weights of the three beacons. We can compute
the position of unknown nodes using weighted least square method, which is proposed in section 3.2.

5. Performance Evaluation

In this section, we first present the performance of the proposed algorithm under secure conditions. After that, we analyze the performance of the proposed algorithm on against compromised unknown sensor nodes and beacon nodes attacks.

5.1. Performance under Secure Conditions

In this section, simulation results are presented and analyzed to evaluate the performance on the sensor localization on the condition there is no any attack. To evaluate the location performance, we compare the DV-Hop and our proposed algorithm.

The performance evaluation of the localization algorithm adopts an average positioning error as an evaluation index, as following formula.

\[ e_i = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \]  

(14)

where \( n \) is the total number of sensor nodes in WSNs, \( N \) is the number of beacon nodes, \((\bar{x}_i, \bar{y}_i)\) is the real coordinate of the unknown node \( i \), \((x_i, y_i)\) is the evaluated coordinate, and \( r \) is the communication range of sensor nodes.

We offer simulation results under the following simulation circumstance: All the sensor nodes are random placed in a square area with the fixed size of 100m \( \times \) 100m. The radio range of sensor nodes \( r \) is set to 15 meter. And there are 20% beacon nodes to all sensor nodes in the network. Figure 1 show the localization error vs. ratio of beacon nodes when the number of sensor nodes is 200. As can be seen from the Figure 1, the proposed weighted DV-Hop algorithm can provide more accurate location estimation than the DV-Hop algorithm. As in figure 1, for the same ratio of beacon nodes, the position error in our weighted DV-Hop algorithm is smaller than in DV-Hop algorithm. For example, the proposed algorithm has an average localization error of about 41% when the beacon nodes ratio is 10%, whereas the DV-Hop has an average localization error of about 48.5%.

![Figure 1. Localization Error vs. Ratio of Beacon Nodes with 200 Sensor Nodes](attachment:figure1.png)
5.2. Performance on Against Compromised Unknown Sensor Nodes and Beacon Nodes

In this section, we evaluate how the proposed scheme improves the network security in terms of resilience against nodes capture attacks.

1) Against Unknown Sensor Nodes Capture Attack

When the attacker compromises an unknown sensor node, he can modify the hop count in the broadcast messages to influence the results of the localization or can use the received message to perform replay attack.

A compromised unknown sensor node will decrease or increase the hop count in the broadcast message. As all sensor nodes in the network has the beacon nodes authentication key, the forged packet with smaller hop counts than the actual count will be detected by the neighbors. In the localization algorithm, a sensor node process the packet with smaller hop count, the forged packet with greater hop count than the actual count have no influence on the location result.

There are two scenarios when malicious unknown sensor nodes perform replay attacks. In the first scenario, the malicious unknown sensor node reply the received the packets broadcasted by its neighbors locally. In this case, it can make the sensor nodes receiving the hop count less one that actual count. And this attack have little affect on the result of localization. In the other scenario, a malicious unknown node tunnels the received broadcast packets to another malicious unknown node at another location and retransmits the packets there to perform wormhole attack. In this case, if the replay place is not far away from the original place, the attack has little effect on the result of localization. If replay place is far away from the original place, the intrusion detection scheme on the sink node will detect the attack.

2) Against Beacon Nodes Capture Attack

When an attacker captured beacon nodes, he can modify the initial hop count, its position, the hop size to influence the results of the localization. Here, we use the simulation to demonstrate the security performance of our scheme. To show the performance of the scheme against the attack, we give the following defines to identity the attack strength.

Definition 1: $B_h$, the attack strength based on forge hop count, $B_h$ is defined as the initial hop count set by the malicious beacon nodes.

Definition 2: $B_p$, the attack strength based on forge position, $B_p = \frac{|P_r - P_f|}{r}$, here $P_r$ is real position of the malicious beacon nodes, $P_f$ is the forged position.

Definition 3: $B_s$, the attack strength based on the forge hop size, $B_s = \frac{d_f - d_r}{d_r}$, here $d_f$ is the real hop size, $d_r$ is the forged hop size.

We offer simulation results under the following simulation circumstance: All the sensor nodes are random placed in a square area with the fixed size of $100m \times 100m$. The radio range of sensor nodes $r$ is set to 15 meter. And there are 20% beacon nodes to all sensor nodes in the network.

Here three graphs of simulation results are presented. Figure 2 show the localization error vs. ratio of malicious beacon nodes when the attackers change the initial hop count to influence the result of localization, the Figure 3 shows that when the attackers change the position of beacon nodes to attack the localization scheme, and Figure 4 shows that when attackers change the hop size to attack the localization scheme. As can be seen from the figures, the proposed can mitigate the
impact of attack effectively. Though the localization error will increased when the attack strength becomes strong, the intrusion detection scheme on the sink node can detect such attack easily.

Figure 2. Performances Against the Forge Hop Count Attack

Figure 3. Performances Against the Forge Position Attack

Figure 4. Performances against the Forge Hop Size Attack
6. Conclusion

In this paper, we presented secure localization scheme to detect and resist the attack for the DV-Hop localization process by giving the weight of beacon nodes. In the scheme, the beacon node is given a weight, which reflects its affect on the position of the unknown sensor nodes. We also conduct analysis and simulations to show the effectiveness of our proposed scheme under different attack and the performance of localization. The performance evaluation results demonstrate that the proposed algorithm can improve localization accuracy and against the nodes capture attacks effectively.

References
