

# Bilateration: An Attack-Resistant Localization Algorithm of Wireless Sensor Network\*

Xin Li<sup>1</sup>, Bei Hua<sup>2,\*\*</sup>, Yi Shang<sup>3</sup>, Yan Guo<sup>4</sup>, and LiHua Yue<sup>5</sup>

<sup>1,2,4,5</sup>Department of Computer Science and Technology  
University of Science and Technology of China, Hefei 230027, China  
<sup>1,2,4</sup>Mobile Computing Laboratory  
Suzhou Institute for Advanced Study, Suzhou, Jiangsu 215123, China  
{xinxin1, guoyan6}@mail.ustc.edu.cn, {bhua, llyue}@ustc.edu.cn  
<sup>3</sup>Department of Computer Science  
University of Missouri-Columbia, Columbia, MO 65211, USA  
shangy@missouri.edu

**Abstract.** Most of the state-of-the-art localization algorithms in wireless sensor networks (WSNs) are vulnerable to attacks from malicious or compromised network nodes, whereas the secure localization schemes proposed so far are too complex to be applied to power constrained WSNs. This paper provides a novel secure scheme “Bilateration” which is derived from multilateration but can be calculated more accurately and quickly to resolve the positions of unknown nodes without explicitly distinguishing what kind of location attacks the WSN is facing. This paper also compares Bilateration with three existing multilateration solutions that optimize the location estimation accuracy via LS, LMS and LLMS respectively in a simulated threat environment. The experiment results show that Bilateration gets the best tradeoff among estimation error, filtering ability and computational complexity.

**Keywords:** localization, WSNs, multilateration, Bilateration, LS.

## 1 Introduction and Related Work

A WSN may run in a hostile environment without any supervision, where the attackers may easily threaten the functionality of position-aware applications by exploiting the vulnerabilities of the localization schemes. There are mainly two types of attacks aiming at the localization process in WSNs [2]. The first type is launched by malicious nodes that are not a part of the network and controlled by an attacker. Typical attacks include modifying distance, jamming communication and creating wormholes [3] [4] in the network. The second type is launched by compromised nodes that are a part of the network and can authenticate

---

\* This work was supported by the National Natural Science Foundation of China under Grant No.60673173 and No.60673111, and the Fund for Foreign Scholars in University Research and Teaching Programs.

\*\* To whom correspondence should be addressed.

themselves as honest nodes, but are controlled by an attacker. They report false positions and disseminate false topology information. Most of the existing localization algorithms don't have the ability to filter out incorrect information, thus are vulnerable to various location attacks.

Recently, some secure localization schemes have been proposed to resist the attacks launched by compromised or malicious nodes. The most common techniques include location-verification[5], distance-verification[6][7], distance-bounding plus some symmetric key cryptography[8], RSS measurements[10] and "packet leashes"[4]. However these methods always require powerful calculation ability, precise synchronization, fast transmission, or somewhat training, etc, which are not suitable for such tiny, low-cost, power constrained sensor nodes. Alternatively, localization based on least median squares (LMS)[9] has been introduced to improve the resilience and accuracy of localization, which however is not suitable to WSN as well due to the heavy burden of calculation. Then [9] chooses to formulate a linearization of the LS (LLMS) location estimator in order to reduce the computational complexity of LMS at the cost of accuracy.

The main contributions of our work are in two aspects. Firstly, unlike the traditional secure localizations which introduce countermeasures to every possible attack, we propose a novel secure localization mechanism Bilateralation, which is efficient in calculation and independent of the type of attacks. Secondly, we compare the performance of Bilateralation with three multilateralation solutions using LS, LMS, LLMS in the simulated settings, and the results show that our method outperforms the other secure schemes in estimation accuracy, filtering ability and computational complexity.

The remainder of this paper is organized as follows. Section 2 formulates the secure localization problem; section 3 overviews the basic idea of LS, LMS and LLMS; section 4 describes Bilateralation algorithm; section 5 compares the performance of the above four algorithms; section 6 concludes the paper.

## 2 Problem Formulation

We consider a homogeneous network consisting of a set of wireless sensor nodes, including anchors and unknown nodes. Sensor nodes are equipped with radio transceivers, and two nodes can communicate with each other if the distance between them is within node's radio range. Each node can measure the distance to other nodes through TDOA, RSSI or something like DV-HOP with a white Gaussian noise:

$$d_{measured}=d_{real}+noise, \text{ noise}\sim N(0,VD) \quad (1)$$

The most remarkable feature of Bilateralation is that it only cares about the result of attack, i.e., the received distance and/or reference location might be false, but needs not distinguish who launches the attack or what kind of attack it is. To simply the description, in this paper we use compromised nodes to refer to both malicious nodes and compromised nodes that issue false distance and/or

reference location information in the network. The threat model we consider is as follows: a few anchor nodes have been compromised and purposely disseminate randomly false reference positions; moreover compromised nodes may not be detected by other means.

Suppose an unknown node located at  $(x_0, y_0)$  has collected a set of reference positions  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  and measured distances  $\{d_1, \dots, d_n\}$  to these nodes. In an idealistic environment setting without any noise and threat, these positions and distances satisfy the following  $n$  equations: [4mm]

$$\begin{aligned} (x_1 - x_0)^2 + (y_1 - y_0)^2 &= d_1^2 \\ &\vdots \\ (x_n - x_0)^2 + (y_n - y_0)^2 &= d_n^2 \end{aligned} \tag{2}$$

If  $n \geq 3$ ,  $(x_0, y_0)$  can be uniquely determined by solving any 3 of the equations if the selected anchors are not in a line. This method is the classical trilateration algorithm. In a 2D plane, solution of trilateration is the intersection of three circles centered at three anchors (fig.1(a)), and solution of multilateration (2) is the intersection of all  $n$  circles (fig.1(b)). However, in a real noisy environment with imprecise reference location and/or distance measurements,  $n$  circles do not intersect at one point. Therefore an objective function (3) is used to minimize the error between estimated position and real position.

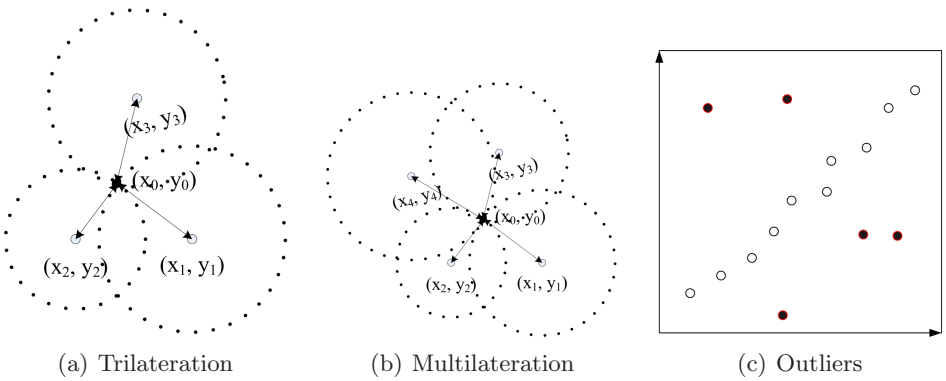


Fig. 1. Multilateration and Outliers

Generally speaking, in a noisy environment without compromised nodes, multilateration with LS is not a bad choice. However, in an environment with some compromised nodes, LS is not good since the estimated position  $(\hat{x}_0, \hat{y}_0)$  may be adversely “removed” far away from the optimal position by compromised nodes. In order to get rid of the “outliers” (see fig.1(c)) caused by compromised nodes, LMS and linear LMS are applied to replace LS in estimation process.

### 3 LS, LMS and LLMS

#### 3.1 Least Square

Multilateration with LS is to minimize the difference between the estimated position  $(\hat{x}_0, \hat{y}_0)$  and the real position  $(x_0, y_0)$  of a node, see (3).

$$(\hat{x}_0, \hat{y}_0) = \underset{(x_0, y_0)}{\operatorname{arg\,min}} \sum_{i=1}^n [\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - d_i]^2 \tag{3}$$

This method usually involves some kind of iterative searching technique such as gradient descent or Newton method. To avoid local minimum LS must run several times with different initial starting points, which is expensive in terms of computing overhead. Moreover, it is vulnerable in the presence of compromised nodes, e.g., if an unknown node receives a false position sent by a compromised node, the estimated position may deviate significantly from its true value even if the measured distance is accurate. This is because pure LS tries to achieve a global optimality of all samples including outliers.

#### 3.2 Least Median Square

To increase the resilience of multilateration with LS, least median squares (LMS) is proposed in [9]. Instead of minimizing the summation of the residue squares, LMS tries to minimize the median of the residue squares:

$$(\hat{x}_0, \hat{y}_0) = \underset{(x_0, y_0)}{\operatorname{arg\,min}} \operatorname{med}_i [\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - d_i]^2 \tag{4}$$

According to [9], the procedure for implementing the robust LMS algorithm is summarized as follows:

1. Set  $n=4$  as the appropriate subset size.
2. Set  $M = \begin{cases} 20, & \text{if } n > 6 \\ \binom{n}{4}, & \text{otherwise} \end{cases}$  as the appropriate total number of subsets.
3. Randomly draw  $M$  subsets of size  $n$  from the heard anchors  $\{(x_0, y_0), \dots, (x_n, y_n)\}$ . Calculate the estimation  $(\hat{x}_0, \hat{y}_0)_j$  using LS for each subset and the corresponding median of residues  $\{r_{ij}^2\}$  for every  $(\hat{x}_0, \hat{y}_0)_j$ . Here  $i=1, 2, \dots, n$  is the index for heard anchors, while  $j=1, 2, \dots, M$  is the index for the subsets.
4. Set  $m = \operatorname{arg\,min}_j \operatorname{med}_i \{r_{ij}^2\}$ , then  $(\hat{x}_0, \hat{y}_0)_m$  is the subset estimation with the least median of residues, and  $\{r_{im}\}$  is the corresponding residues.
5. Calculate  $s_0 = 1.4826(1 + \frac{5}{n-2}) \sqrt{\operatorname{med}_i r_{im}^2}$ .
6. Assign weight  $\omega_i$  to each heard positions with equation  $\omega_i = \begin{cases} 1, & \left| \frac{r_i}{s_0} \right| \leq \lambda \\ 0, & \text{otherwise} \end{cases}$ .
7. Do LS to all heard positions with weights  $\{\omega_i\}$  to get the final  $(\hat{x}_0, \hat{y}_0)$ .

### 3.3 Linear LMS

Considering that finding estimation for M subsets requires a lot of computation, [9] transforms nonlinear LS into linear LS, which is a suboptimal but more computationally efficient algorithm.

1. Average all the left parts and right parts of (2):

$$\frac{1}{n} \sum_{i=1}^n [(x_i - x_0)^2 + (y_i - y_0)^2] = \frac{1}{n} \sum_{i=1}^n d_i^2 \tag{5}$$

2. Subtract (5) from each equation in (2), and linearizes to get the following new equations:

$$\begin{aligned} (x_1 - \frac{1}{n} \sum_{i=1}^n x_i)x_0 + (y_1 - \frac{1}{n} \sum_{i=1}^n y_i)y_0 &= \frac{1}{2}(x_1^2 + y_1^2 - d_1^2 - \frac{1}{n} \sum_{i=1}^n (x_i^2 + y_i^2 - d_i^2)) \\ &\vdots \\ (x_n - \frac{1}{n} \sum_{i=1}^n x_i)x_0 + (y_n - \frac{1}{n} \sum_{i=1}^n y_i)y_0 &= \frac{1}{2}(x_n^2 + y_n^2 - d_n^2 - \frac{1}{n} \sum_{i=1}^n (x_i^2 + y_i^2 - d_i^2)) \end{aligned} \tag{6}$$

3. Estimate  $(\hat{x}_0, \hat{y}_0)$  by linear least squares.

Transforming nonlinear LS into linear LS saves much computation, since the solution can be calculated directly from (6) without iterative searching and repeating. Furthermore, the solution of linear LS can be used as the starting point of nonlinear LS to prevent nonlinear LS from getting trapped in a local minimum. In our simulation, we use this starting point to do nonlinear LS.

However, due to the subtraction, the optimal solution of linear equations in (6) is not exactly the same as that of nonlinear LS in (2), which means much accuracy is lost, especially when the number of heard anchors is small. In the experiments of [9], as the number of heard anchors is fixed to 30, linear LS is acceptable as it still performs very well. However, 30 heard anchors per unknown node is not practical in the realistic settings.

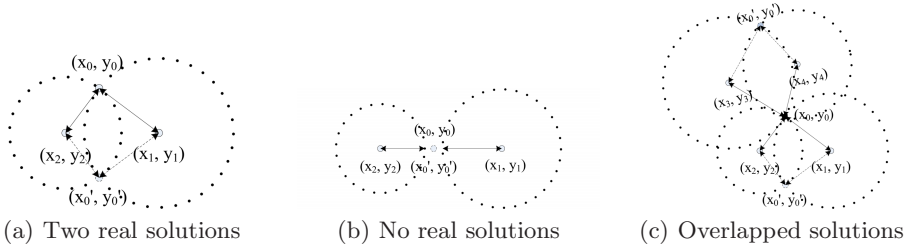
## 4 Bilateration

Due to inherent limitation, the performance of LMS and LLMS is poor when the number of heard anchors is small, or the percentage of outliers exceeds 50% even if there are still many usable samples. The goal of bilateration is to achieve the same accuracy as LMS and the same computational speed as LLMS, meanwhile its performance is less affected by the number of heard anchors and the percentage of outliers.

In an idealistic environment without measurement noise and attacks, if  $n=2$  we can solve (2) as follows:

$$\begin{aligned}
 x_0 &= \frac{-(mn - ny_1 - x_1)}{1 + n^2} \pm \frac{\sqrt{2(nx_1 + m)y_1 - y_1^2 - n^2x_1^2 - 2mnx_1 - m^2 + (1 + n^2)d_1^2}}{1 + n^2} \\
 y_0 &= m + nx_0 \\
 m &= \frac{1}{2} \frac{(x_1^2 - x_2^2) + (y_1^2 - y_2^2) - (d_1^2 - d_2^2)}{y_1 - y_2} \\
 n &= -\frac{x_1 - x_2}{y_1 - y_2}
 \end{aligned} \tag{7}$$

Evaluation of (7) is very fast given the value of  $\{(x_1, y_1), (x_2, y_2), d_1, d_2\}$ . The real solutions for  $(x_0, y_0)$  are called **candidate positions**, which in a 2D plane are the intersections of two circles (see fig.2(a)); the complex solutions are not considered in this paper (see fig.2(b)). If another two anchors (at least one of the them doesn't belong to  $\{(x_1, y_1), (x_2, y_2)\}$ ) and corresponding distances are selected, another two candidate positions are solved for  $(x_0, y_0)$ . Among the 4 candidate positions, at least 2 positions overlap each other, and this overlapped point is the correct solution for  $(x_0, y_0)$  (see fig. 2(c)). If more anchor positions and distances are available, more overlapped points will occur. In this way, even most of the heard anchors are compromised, this method can correctly locate an unknown node as long as at least three anchors and corresponding distance measurements are accurate.



**Fig. 2.** Bilateralation

In real noisy environment, there may be no overlapped points due to distance error. However, there is reason to believe that correct positions should be close to each other if the distance error is bounded. We define:

**Correct candidate positions:** a group of candidate positions, in which there is at least one position whose distances to the other members are less than the threshold  $\delta$ .

**Candidate neighbors:** two candidate positions between which the distance is within  $\delta$ .

For an unknown node  $\mu$ , the procedure for implementing our Bilateralation algorithm is summarized as follows:

1. If  $n \leq 3$ , set  $\mu$  as un-localized and terminate the algorithm. This situation will not be considered in our performance comparison, because there is no way to distinguish which position is false.

	$\mu$	$n_1$	$n_2$	$n_3$							
$a_1$	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0	=	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1
1											
0											
0											
0											
1											
$a_2$	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0		<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1
1											
0											
0											
0											
1											
$a_3$	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1		<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>4</td></tr></table>	4
1											
1											
1											
1											
4											
$a_4$	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0		<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>3</td></tr></table>	3
1											
1											
1											
0											
3											
$a_5$	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr></table>	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td></tr></table>	0		<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td></tr></table>	2
1											
0											
1											
0											
2											

$\langle ?(1+1+4+3+2)/5 \rangle$ 

t
t
f
f
t

$\mu$  collects three weight tables from its neighbors  $n_i$  ( $i=1, 2, 3$ ), which records the weight of 5 anchors  $a_j$  ( $j=1, \dots, 5$ ); t or f represents true or false compromised node.

**Fig. 3.** Who is the compromised node

2. Exhaust all the combinations of two anchors and the corresponding measured distances  $\{(a_i, d_i), (a_j, d_j)\}$  to evaluate (7), and suppose  $M$  candidate positions  $\{c_1, \dots, c_M\}$  have been solved from the  $\binom{n}{2}$  combinations.
3. For each candidate position  $c_i$ , calculate  $\{D_{i1}, \dots, D_{ii-1}, D_{ii+1}, \dots, D_{iM}\}$ , where  $D_{ij}$  is the distance between  $c_i$  and  $c_j$ ,  $i, j=1, 2, \dots, M$  is the index to candidate positions.
4. For each  $c_i$ , find out all the distances shorter than the threshold  $\delta$  and get  $\{D_{ip}, \dots, D_{it} | D_{ip} < \delta \wedge \dots \wedge D_{it} < \delta, D_{ip}, \dots, D_{it} \in \{D_{i1}, \dots, D_{ii-1}, D_{ii+1}, \dots, D_{iM}\}\}$ , set  $n_i = |D_{ip}, \dots, D_{it}|$ . ( $|\cdot|$  denotes the cardinality of a set).
5. Find out  $m = \text{argmax}_i \{n_i\}$ , suppose  $\{D_{mp}, \dots, D_{mt}\}$  are the distances between  $c_m$  and its candidate neighbors  $\{c_p, \dots, c_t\}$ ; find out the corresponding anchors  $\{a_l, \dots, a_q\} \subseteq \{a_1, \dots, a_n\}$  from which  $\{c_m, c_p, \dots, c_t\}$  are solved; set the weights of  $\{a_l, \dots, a_q\}$  as 1; set the weights of the other heard anchors as -1.
6. Exchange the weight table with its neighbors.
7. Collect all the weight tables from its neighbors; pick out the common heard anchors; add their weights together; set the anchors whose weight is less than the average weight as the compromised nodes. (see fig.3)
8. Delete the candidate positions caused by compromised nodes from  $\{c_1, \dots, c_M\}$ ; set the average of all the left candidate positions as the final estimated position  $e_\mu$ .

If the unknown node hears 4 different positions including 1 false position, LMS and LLMS are unable to deal with this situation, whereas our scheme can find out the correct positions if the distance between the correct candidate positions is shorter than  $\delta$ .

## 5 Simulation

To evaluate Bilateration, we simulated it and multilateration with LS, LMS, and LLMS (abbr. to LS, LMS and LLMS) on Matlab, and compared them in terms of estimation error, ability of false position filtering, and computational complexity in simulation environment. Estimation error is the average variance between estimated locations and real locations. Ability of false position filtering is the average number of false positions that are used in the location estimation of each unknown node. Each data point represents the average value of 500 trials

with different random seeds. We use ideal LS as a benchmark in the performance comparisons, which can filter out all the compromised anchors before estimation.

In our simulation settings, we have the following definitions and assumptions.

- Anchors and unknown nodes are uniformly distributed in an area of  $200 \times 200m^2$ .
- The coordinates of false positions,  $x$  and  $y$ , are independent and identically follow normal distribution  $N(100, VP)$ , where  $VP$  varies from 20 to 200m.
- The noise of measured distance obeys normal distribution  $N(0, VD)$ , where  $VD$  varies from 0 to 50m.
- $R$  is the radio range of node, and is fixed to 50m in our experiments.
- $NA$  is the average number of heard anchors by each unknown node.
- $NU$  is the average number of heard unknown nodes by each unknown node.
- $CP$  is the percentage of compromised anchors, and varies from 0 to 1.

In the following experiments, without specification, the default environment settings are:  $VP=20m$ ,  $VD=5m$ ,  $NA=7.5$ ,  $NU=7.5$  and  $CP=0.2$ . After a lot of experiments with different  $\delta$  and  $\lambda$ , we find out that the optimal  $\delta$  for Bilateralation is 10, and the optimal  $\lambda$  for both LMS and LLMS is 1.5. We omit the detailed performance comparison due to limitation of space.

### 5.1 Influence of Average Number of Anchors

In this experiment, we investigate the influence of average number of heard anchors ( $NA$ ) on the performance of the four localization algorithms.

In fig.4(a), except for LS whose estimation error increases about 5% when  $NA$  increases from 5 to 25 due to its lack of filtering ability, the estimation error of other four algorithms (including Ideal LS) decreases. Bilateralation has lower estimation error than LMS and LLMS, but their gap shrinks when  $NA$  increases.

Fig.4(b) compares the filtering ability of all algorithms. Since LS doesn't filter out outliers and  $CP$  is fixed, the false positions used by LS increases with  $NA$ . The number of unfiltered false positions used by Bilateralation is smaller than that used by LMS and LLMS; that is to say, Bilateralation has stronger filtering ability

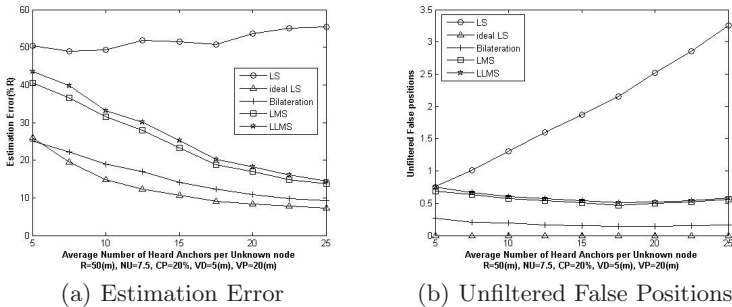


Fig. 4. Influence of Average Number of Anchors



than LMS and LLMS. This explains why Bilateration has lower estimation error than LMS in a hostile environment: the stronger filtering ability compensates for the suboptimal estimation accuracy.

In fig.4(a), the estimation error of Bilateration is close to that of Ideal LS all the time, whereas LMS and LLMS requires many more anchors to get the same accuracy. Since there are only a few anchors in a real wireless sensor network, this result shows that Bilateration is more suitable to real settings.

### 5.2 Influence of Percentage of Compromised Nodes

In this experiment, we investigate the influence of compromised percentage (CP) on the performance of algorithms. It is interesting to observe that ideal LS terminates when CP reaches 0.6, this is because the number of un-compromised anchors is smaller than 3 for each unknown nodes with  $NA=7.5$ . Therefore it is meaningless to discuss the performance for CP larger than 0.6.

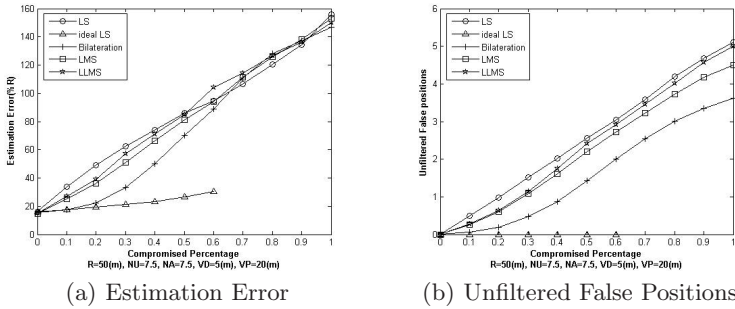


Fig. 5. Influence of Percentage of Compromised Nodes

In fig.5(a), the estimation error of Bilateration is lower than that of LS, LMS and LLMS all the time when CP is smaller than 0.6, which shows that Bilateration is less affected by CP. However the four curves tend to approach when CP increases, since there is no difference among them when no right position is available.

In fig.5(b), the number of unfiltered false positions used by Bilateration is smaller than that used by LS, LMS and LLMS, which shows that Bilateration has the strongest filtering ability.

### 5.3 Influence of Distance Measurement Error

In this experiment, we investigate the influence on distance measurement error on the performance of algorithms.

In fig.6(a), the estimation error of Bilateration increases rapidly as the variance of distance (VD) increases. The estimation error of Bilateration is lower than that of LS, LMS and LLMS when VD is less than 13, then it exceeds them

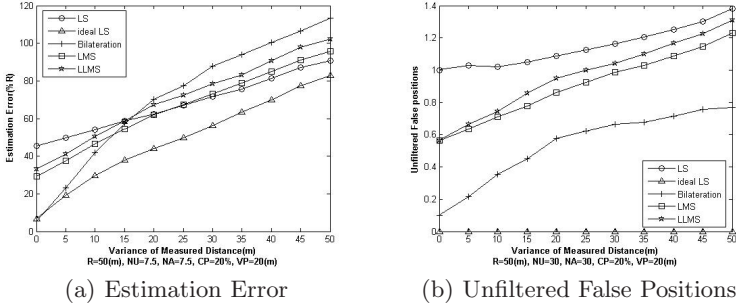


Fig. 6. Influence of Distance Measurement Error

quickly. We observe that the estimation error of Bilateration does not reach 0 even when VD is 0, since  $\delta=10$  allows some false positions to participate in the location estimation (fig.6(b)). If  $\delta$  is set to 0, then Bilateration can filter out all the false positions when VD is 0. LMS and LLMS outperform LS when VD is less than 22 and 15 respectively, and then lost their advantage as well.

In fig.6(b), the number of unfiltered false positions used by all the four algorithms increase as VD increase, since large distance error makes it more difficult to distinguish between correct position and false position. Therefore if the measured distance is not accurate enough, the filtering ability of algorithm has no meaning.

This experiment shows that Bilateration is more suitable to work in an environment with moderate noise that is less than 24% or radio range.

### 5.4 Tradeoff Between Performance and Communication Complexity

Bilateration is the only algorithm which needs to communicate with heard unknown nodes to identify the compromised nodes. However, the performance of Bilateration with small NA is not sensitive to the average number of unknown nodes. Meanwhile, the performance of the other three algorithms are not sensitive to the average number of unknown nodes (Fig.7(a)(b)). So, in this experiment, we only evaluate Bilateration with  $NA=25$  and different CPs.

Fig.7(c)(d) evaluate Bilateration with different CPs. The estimation error and unfiltered false positions of Bilateration with bigger CP decreases more rapidly than those with smaller CP as NU increases. So the strategy of exchanging weight tables is more useful for Bilateration with big NA and CP. In other words, if CP or NA is not big we can abandon this strategy to save energy.

### 5.5 Computation Complexity Analysis

Since Bilateration, LMS and LLMS primarily differ in the means of estimation, we only analyze the amount of computation involved in estimation. Suppose that an unknown node  $\mu$  hears  $n$  anchors.

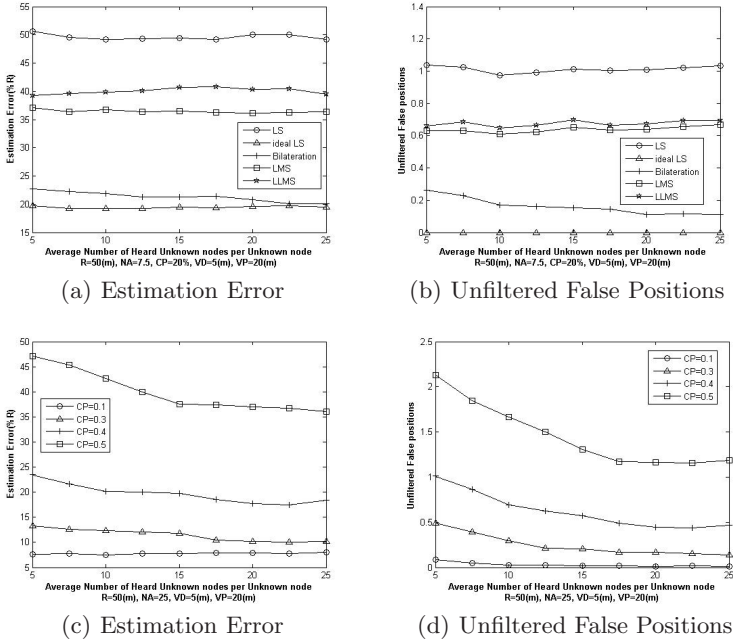


Fig. 7. Influence of Average Number of Unknown nodes

In LMS,  $\mu$  needs to do LS estimation  $\binom{n}{4}+1$  times, when  $n \leq 6$  or 21 times when  $n > 6$ . In each round of LS estimation except for the last round, four anchor positions are involved in the estimation calculation, and in the last round all the unfiltered positions are involved. In order to avoid local minimum, solution of linear LS is used as a start point to search the global optimality, which adds another  $\binom{n}{4}+1$  or 21 times of linear LS calculation. It is possible to use LS without preliminary linear LS, however, LS may need to search more times for optimal solution from different start point, and moreover the solution may be trapped into local minimum.

In LLMS,  $\mu$  needs to do linear LS  $\binom{n}{4}+1$  times when  $n \leq 6$  or 21 times when  $n > 6$ . The amount of computation involved in linear LS is much less than that involved in LS.

In Bilateration,  $\mu$  needs to evaluate (7)  $\binom{n}{2}$  times to find out all the candidate positions, and then perform  $4\binom{n}{2}^2$  times of distance calculation between each candidate position to every other candidate position. All the computation only involves simple algebraic calculation, so Bilateration runs much faster than LMS and comparable with LLMS, which was verified by our experiments as well.

## 6 Conclusion and Future Work

In this paper we propose Bilateration, an attack-resistant localization algorithm that tries to find a set of close-by positions from all candidate positions and use

the average of these close-by positions as the estimated position. Bilateralation is resilient to all kinds of position and distance cheating attacks in the sense that it only cares about the result of attacks rather than the process of attacks; and is more close to the real world.

In the threat model of this paper, we assume that compromised nodes do not cooperate and disseminate randomly false positions. If some or all of the compromised nodes cooperate to give false but specious positions, e.g., each compromised node reports a position that is a fixed displacement to its real position, then detecting and filtering these nodes is difficult. In the future, we will evaluate Bilateralation and other three localization algorithms under this kind of attack.

## References

1. Lymberopoulos, D., Lindsey, Q.: An Empirical Analysis of Radio Signal Strength Variability in IEEE 802.15.4 Networks using Monopole Antennas, ENALAB Technical Report 050501, Yale University (2006)
2. Capkun, S., Hubaux, J.P.: Secure positioning in sensor networks, Technical report (May 2004)
3. Xu, W., Wood, T., Trappe, W., Zhang, Y.: Channel surfing and spatial retreats: defenses against wireless denial of service. In: Proceedings of the 2004 ACM workshop on Wireless security, pp. 80–89 (2004)
4. Hu, Y.C., Perrig, A., Johnson, D.: Packet leashes: a defense against wormhole attacks in wireless networks. In: Proceedings of IEEE Infocom, pp. 1976–1986 (2003)
5. Brands, S., Chaum, D.: Distance-bounding protocols. Theory and Application of Cryptographic Techniques, 344–359 (1993)
6. Sastry, N., Shankar, U., Wagner, D.: Secure Verification of Location Claims. In: Proceedings of WiSe (2003)
7. Waters, B., Felten, E.: Proving the Location of Tamper-Resistant Devices. Technical report, Princeton University
8. Čapkun, S., Buttyán, L., Hubaux, J.-P.: SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In: Proceedings of SASN (2003)
9. Li, Z., Trappe, W., Zhang, Y., Badri Nath.: Robust statistical methods for securing wireless localization in sensor networks. In: Proceedings of IPSN (2005)
10. Kusy, B., et al.: Node-density independent localization. In: IPSN (2006)
11. Zhou, G., et al.: Impact of Radio Irregularity on Wireless Sensor Networks. In: Proceedings of 2nd MobiSys (2004)