Secure Hop-Count Based Localization in Wireless Sensor Networks

Yingpei Zeng†, Shigeng Zhang†,
†State Key Laboratory of Novel Software Technology
Nanjing University
Nanjing 210093, P.R.China
{zyp,zsg}@dislab.nju.edu.cn

Shanqing Guo‡, Li Xie†
‡School of Computer Science and Technology
Shandong University
Jinan 250100, P.R.China
guoshanqing@gmail.com,xieli@nju.edu.cn

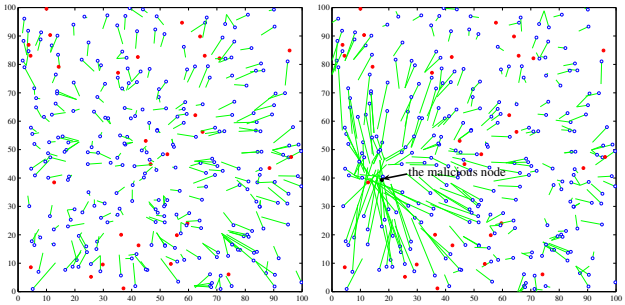
Abstract

*Many sensor network applications rely on sensors' location information. However, most of existing location algorithms assume a non-adversarial environment or assume beacons lying within 1-hop. In this paper, we focus on Hop-Count based localization (multihop) and develop a **S**ecure **HO**p-Count based **LOC**alization scheme, called SHOLOC, to make localization attack-resistant. In SHOLOC, we assume both ordinary nodes and beacon nodes can be moving, and the whole network periodically restarts localization. SHOLOC proposes a protocol to authenticate beacon information and protect hop-count from being arbitrary changed. SHOLOC employs beacon nodes to detect wormhole attacks. Theoretical analysis and simulation results are presented. Our conclusion also includes two interest findings: 1)hop-count increment attacks are not effective to Hop-Count based algorithms, 2)filter mechanisms such as least median squares (LMS) are not resistant to wormhole attacks, and our method of detecting at beacon nodes side works well.*

1. Introduction

Determining the location of sensors reliably is an important issue in wireless sensor networks (WSN). On one side, many applications of WSN require knowing location of sensors. e.g., collected data need to be bound with location, and location information also can be used to facilitate network functions such as routing. Also there are many security mechanisms employing sensors' location recently, e.g., key distribution [9]. On the other side, when sensor networks are deployed in real world, the environments may be untrustworthy, and even may be hostile with presence of malicious adversaries, e.g., in the battlefield-related applications[1]. Secure localization algorithms are needed to survive in such environments.

Several secure localization algorithms have been proposed. Capkun and Hubaux [2] proposed SPINE for verifying distances between nodes in sensor network positioning. Li *et al.* [8] proposed to use robust statistical methods to reduce the effect of bad location references. Liu *et al.* [10] proposed Attack-Resistant MMSE to filter bad location references too. However, to the best of our knowledge, all the existing secure localization proposals didn't consider the *multihop* property. Both [8] and [10] only consider the last location-calculation step, they can't be applied to *multihop* localization algorithms directly.



(a) without malicious nodes (b) one malicious node

Figure 1. Damage of a single malicious node. Red nodes represent beacons, black node is the malicious node, and lines represent the estimation error. The simulation is done in $100\times100m^2$ area, total 300 nodes and 30 (10%) are beacons, radio range is 15m, beacon packets propagation is limited to 6 hops.

Here, we focus on the security of Hop-Count based multihop localization algorithms [12][15][11], for their simplicity and efficiency. Hop-Count based localization algorithms usually have little assumption about the ability of sensor node, and usually can determine sensor location with location error about 30% of the radio range, which is sufficient for applications like position-based routing. If more information such as distance level is available, location error can be further reduced to 10% of radio range [11] [15]. However, this class of localization algorithm is vulnerable to attacks. In Figure 1(b), there is only a single malicious node, which reduces the hop count of pass-by packets by 3, we can see that the location estimation of nearby nodes are severely distorted.

In this paper, we present a Secure HOp-Count based LOCalization scheme (SHOLOC), which is resistant to different

attacks, e.g., hop-count reduction attack and forging packets. A protocol combining modified TESLA [13] and hash mechanism is proposed to authenticate beacon location information and protect hop-count information. In order to detect wormhole attacks, we propose a method which employing beacon nodes to check the distance-impossibility between them. Finally, we use least median squares (LMS) to cope with minor bad location references. Theoretical analysis and simulation results are also presented.

The rest of the paper is organized as follows. The next section summarizes similar efforts in current research, Section 3 describes network model and attacker model. In Section 4 we present the SHOLOC scheme and we give analysis in the Section 5, We give simulation results in Section 6. Section 7 presents our conclusion.

2. Related Work

2.1. Hop-Count based Localization in Ad Hoc and Sensor Networks

Many different localization algorithms have been proposed [16][12][15][11] [17]. Hop-Count based localization algorithms [12][15][11] are a type of simple but efficient algorithms. Niculescu and Nath first proposed DV-Hop in [12]. Savarese and Rabaey [15] proposed Hop-TERRAIN which contains a refinement process. Nagpal *et al.*[11] proposed a similar coordinate system and showed more theoretical results. In [17], Wang and Xiao aimed to adapt Hop-Count localization to concave areas.

Hop-Count based localization algorithms usually work as follows. First, each beacon propagates a packet containing its location and a hop-count field, each neighbor node who doesn't hear that packet before will increase the *hop-count* field by one, and then forwards it to neighbors, if the packet hasn't reached propagation hop-limit. After that, beacons calculate the *corrections* (average length of one hop)using other beacons' locations and hop counts between them, and propagate corrections to the network again. At last, each ordinary node has several (need at least 3 when is in 2D) beacons' locations and corresponding distances to them $< loc, dist >$, which are usually called location references, where $dist = hopcount * mean(corrections)$. Node can then calculate its location using minimum mean square estimate (MMSE)[16] or other mechanisms like bounding-box.

2.2. Secure Localization Algorithms

Recently, secure localization is gaining more and more attention. In [7], Lazos and Poovendran proposed a range-independent localization algorithm called SeRLoc, in which there are locators equipped with sectored antennas, and sensors determine their location by calculating the center of the intersection of heard sectors. Capkun and Hubaux [2] proposed SPINE for sensor network secure positioning. They use distance bounding to verify distances between nodes, and use a proposed mechanism called Verifiable Multilateration (VM) to restrict the possible location of nodes. Both SeRLoc and SPINE need to equip special hardware. Li *et al.* [8] proposed to use robust statistical methods, i.e., least median of squares (LMS) instead of least squares (LS, or alias MMSE), to reduce the effect of bad location references. Liu *et al.* [10] did similar thing with [8], they proposed Attack-Resistant MMSE which limits the mean square error to filter bad location references. Both [8] and [10] only consider the last location-calculation step; they can't be applied to multihop localization algorithms directly.

Our SHOLOC scheme considers both information propagation and location calculation. We will show later that filter mechanisms like [8] and [10] which only considering location calculation step, will not survive in wormhole attacks, because bad location references are dominant in this case. Also, SHOLOC doesn't require to equip special hardware, and use TESLA[13] which is energy efficient to authenticate broadcast packets.

3. Model

3.1. Network Model

The network consists of a set of N sensor nodes, and part of them are beacons (N_{bea}). Beacons' position can be acquired through GPS. Only beacons know their location, since equipping every node with a GPS receiver is usually not feasible due to cost and form factor limitations. Each node has an identifier.

We assume that all the network nodes are deployed randomly in a specific network region of area (S_{area}), and both beacon nodes and ordinary nodes can move randomly in it. To make ordinary nodes know their location, the whole network periodically restarts localization.

To use TESLA, we assume that nodes are loosely synchronized. Beacon nodes can easily be synchronized if they are equipped with GPS. Ordinary nodes can synchronize to their nearest beacons. Also, we assume that each node has the initial disclose key (end of their computed hashchain [13]) of all beacon nodes. e.g., if there are 50 beacons, the hash output of SHA1 is 160-bit, then each sensor only has a storage overhead of 1k bytes.

3.2. Attacker Model

We assume that attackers can be *internal* or *external*, the difference between them is whether they are valid nodes in the network. Internal attackers have more information, they can modify fields of beacons' packets, e.g., reduce the hop-count, which can result in hop-count reduction attacks as show in Figure 1. And they can also increase hopcount fields of pass-by packets more than 1, result in hop-count increment attacks. External attackers can launch wormhole attacks [4], which tunnel and replay packets without need of any information. Both

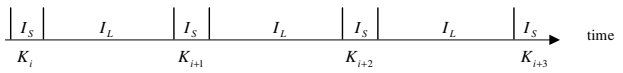


Figure 2. TESLA usage in SHOLOC. The n-length hashchain is produced by hash on a random chose value K_n, initial key is K_0, Keys used in I_S will be disclosed in the followed I_L interval.

types of attacker can change the contents of beacons' packets, if packets don't have integrity protection.

4. SHOLOC: a Secure Hop-Count based Localization scheme

4.1. Beacon Information Authentication

Sensor nodes should be able to authenticate received beacons' packets. Otherwise, any node can forge a packet. Then a question arises: *choose symmetric cryptography or asymmetric cryptography?* Though asymmetric cryptography is indeed becoming feasible to be used in sensor networks, it's still recommended to reduce the usage whenever possible [3]. Also, if we need to verified several packets in a timely manner, asymmetric cryptography still has limitations. e.g., one Elliptic Curve Cryptography (ECC) signature verification takes 1.62s [3], if sensor node receive 20 beacons' packets, it will take 32.4s total, the location information will be too stale to use. TESLA [13] differs from traditional asymmetric protocols in that TESLA achieves asymmetry from clock synchronization and delayed key disclosure, and it only uses efficient symmetric cryptography when authenticating packets.

We propose to use a slightly modified version of TESLA [13], which is shown in Figure 2. In original TESLA, keys are assigned to all time intervals, but in SHOLOC beacons periodically restart localization, and packets are only sent at specified intervals. In order to save many keys elapsed in spare intervals, we propose not to bind keys to these intervals, but only bind to re-localizing interval. We use I_S to represent these short re-localizing interval, and I_L to represent long spare intervals.

4.2. Beacons Information Propagation Protocol

We now describe the protocol used in SHOLOC for location information propagation. Our protocol uses TESLA described in Section 4.1 to authenticate packets, and uses hash function to make the hop-count unreducible and verifiable, which is inspired by secure distance vector route protocol like SEAD [5]. We note here that our protocol can also use alternative techniques (e.g., digital signatures) to authenticate packets.

The protocol is comprised of two phases, as shown in Figure 3.

i) First, all beacons broadcast a BEALOC packet to the network, which contains fields $< BEALOC, bid, loc, seq, t_i, h_0, c, mac, nodelist >$, bid and loc are beacon id and beacon location respectively, seq is the unique sequence number of the packet, t_i is the interval number of TESLA, h_0 is hash result of a random chose value h_r, c is the correction (average length of one hop) of beacon, mac is the message authentication code (MAC) generated by using the key K_i in current interval, $nodelist$ is the path the packet passed. When a node receives a BEALOC packet which is *fresh* to it, it first checks that whether the arrival time $Tr < T_0 + t_i * (I_S + I_L) + d * I_S - \Delta$, where T_0 is the initial time, d is the delay of TESLA which is 1 in our scheme, Δ is the maximum synchronization error between any two nodes. If the time condition holds, it means that the sender hasn't disclose the K_i yet and the packet is valid. *Fresh* means that the node hasn't heard the $< bid, seq >$ pair before or the length of attached $nodelist$ (hop-count) is smaller than the length of same $< bid, seq >$ pair received before (i.e., a shorter path discovered). If the packet is valid, it further checks whether the predecessor has added its id in the $nodelist$, if so it replaces the hash field with $h_j = H(h_{j-1}, selfId)$, attaches $selfId$ in the $nodelist$ field of the packet and forwards it again. Nodes carry on this process until all packets reach MAX_PRO_HOP limit.

ii) Second, when the I_L time interval following I_S arrives, beacons broadcast $AUTH$ packets to disclose the key K_i used in I_S, and also disclose the random chose value h_r. Each node received K_i can verify it by hashing to a key authenticated previously, as doing in TESLA. Then nodes can authenticate the received BEALOC packets by computing MACs of those packets and comparing MACs with their mac fields. Also the hash field can be checked by comparing received h_j with $H(..., H(B, H(A, H(h_r))))$.

Beacon:	$h_0 = H(h_r)$,$mac = MAC_{K_i}(BEALOC, bid, loc, seq, t_i, h_0, c)$
Beacon→ *:	$< BEALOC, bid, loc, seq, t_i, h_0, c, mac, () >$
A:	$h_1 = H[A, h_0]$
A→ *:	$< BEALOC, bid, loc, seq, t_i, h_1, c, mac, (A) >$
B:	$h_2 = H[B, h_1]$
B→ *:	$< BEALOC, bid, loc, seq, t_i, h_2, c, mac, (A, B) >$
...	... (MAX_PRO_HOP)
Beacon→ *:	$< AUTH, bid, seq, t_{i+1}, h_r, K_i >$
A,B,..., forward AUTH packets until reaching MAX_PRO_HOP	

Figure 3. Information propagation in SHOLOC.

When the above protocol ends, each node gets nearby beacons' location information, and the hop-count between them. With the received correction c, each node can get location references in the form $< loc, dist >$. Also, nodes have gotten the shortest paths to nearby beacons as bonus. Then, nodes can use LMS to estimate their locations, see Section 4.4.

Remark 1: Here, the corrections c (average length of one hop) used by each node are the value of last localization round.

Usually, beacon nodes compute the corrections c using received location information of other beacons, so we should send the new calculated corrections c after receiving $AUTH$ packets. But follow our assumption, node randomly moves in the field, and given the total node number isn't changing very fast, the hop length will not change too much, so we can save one packet and use the corrections of last round, which are sent in the BEALOC packets. Simulation results in Section 6.5 also confirm this.

When in the initial round, corrections can by estimated by beacons following formula in [6].

$$d_{hopLen} = r(1 + e^{-n} - \int_{-1}^{1} e^{-\frac{n}{\pi}(arccost - t\sqrt{1-t^2})}dt) \quad (1)$$

where r is radio range, n is the average neighbor number of nodes, which can be computed as $n = \pi r^2 N/S_{area}$[6]. In order to mitigate the effects of paths including malicious nodes, here beacons use *median* instead of *mean* to calculate the corrections.

4.3. Detecting Wormhole Attacks

Wormhole attack [4], is a kind of severe attack in which attackers tunnel packets in one location to another location and then replay them. This kind of attack can succeed without need of any information, and communication authenticity and confidentiality have no effect on it. If a wormhole link presents, all the packets will be tunneled to other side and form bad location references there, and then bad references may be dominant. Filter mechanisms like [8] [10] can be resistant to at most 50% outliers [8]. So only using filter mechanisms like [8] [10] is not resistant to wormhole attacks. This is confirmed by simulation in Section 6.3.

We propose to employ beacon nodes to detect wormhole attacks. Each beacon node B_i, if it has received another beacon node B_j'authenticated location information $< loc_{B_j}, hopcount >$, it then check if:

$$distance(B_i, B_j) > hopcount * r + 2v_{max}(T_s - T_r) \quad (2)$$

where r is the radio range, v_{max} is the maximum possible speed, T_s and T_r are the sending and receival time of B_j's BEALOC packet. Beacons are synchronized so we ignore the clock difference here. If Inequality 2 hold, then beacon node B_i can conclude that the path from B_j to it has a wormhole. Because given the radio range is r, then packets from B_j can't travel $hopcount$ hops to reach B_i, unless there is a wormhole. We can easily see that the false positives is zero, and in Section 5.1, we will analysis the detection ratio.

If wormhole is detected, beacons can report this to base station to take some actions, e.g., locate the wormhole and manually remove it. *We note here that we don't rely on sensor nodes to filter these tunneled location references*, it's the responsibly of the base station, etc.

4.4. Location Estimation Using LMS

We use least median of squares (LMS) to filter bad location references and estimate nodes' location. Since if there are malicious beacon nodes who send wrong location, part of location references will be bad. LMS algorithm was proposed in [14]. Compared with least squares (LS, MMSE) algorithms, LMS is more robust to outliers. Instead of minimizing the summation of the residue squares, LMS minimizes the *median* of the residue squares. Li *et al.* use LMS method to reduce the effect of bad location references in [8]. We choose the LMS-based algorithm instead of another Attack-Resistant MMSE by Liu *et al.* [10], because the latter [10] needs to know the maximum measurement error, which can't be calculated directly in Hop-Count based localization algorithms.

LMS hasn't been studied in multihop localization algorithms like Hop-Count based localization algorithms. We implement the subsamples based LMS ([14], Chapter 5) in location estimation, while the LMS parameters are the same as in [10]. We find that LMS doesn't work well as it supposed to be (50% outliers), fortunately here we only need LMS to defeat quite a few malicious beacons. We give the simulation results in Section 6.2.

5. Analysis

In this Section, we analysis the detection probability of wormhole attacks in SHOLOC, and give a brief security analysis of SHOLOC.

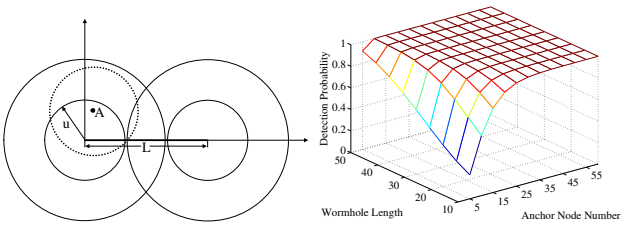
5.1. The Detecting Probability of Wormhole Attacks

For simplicity we don't consider the effect of motion here. We use P_{detect} to represent the detection probability of wormhole attacks. Figure 4(a) illustrates a wormhole with two endpoints. We use $P_{leftDetect}$ and $P_{rightDetect}$ to represent the detection resulted by packets from right endpoint to left endpoint and left endpoint to right endpoint respectively. Because of the symmetry property it's easy to see $P_{leftDetect} = P_{rightDetect}$. According to our detection mechanism in Section 4.3, both left and right beacons should detect the wormhole at the same time, so

$$P_{detect} = P_{leftDetect} \quad (3)$$

We consider $P_{leftDetect}$ with different cirque parts, and there are MAX_PRO_HOP cirques at the left endpoint of the wormhole link. It's difficult to give a close formula to compute the probability, we will show the computation process step by step. Given each cirque detecting independently, we have

$$P_{leftDetect} = 1 - \prod_{i=1}^{MAX\_PRO\_HOP} (1 - P_{left\_i\_cirqueDet}) \quad (4)$$



(a) Analysis of detection of wormhole. At each end point of the L length wormhole link, there are MAX_PRO_HOP circles around (e.g., MAX_PRO_HOP is 2 here), the interval among them is the theoretical Hop Length u.

(b) Demonstrating the detection probability of wormhole attacks. MAX_PRO_HOP is 4 here.

Figure 4. Analysis of wormhole detection.

where, $P_{left\_i\_cirqueDet}$ is the detection probability of the i cirque at left, which can be computed using

$$P_{left\_i\_cirqueDet} = \sum_{j=1}^{N_{bea}} P\_j\_beaIn\_i\_cirque \times (1-(1-P_R)^j)$$

here, $P\_j\_beaIn\_i\_cirque$ is the probability of j beacons lying in the i cirque, P_R is the detection probability of a beacon in i cirque at left.

$$P_R = 1 - \prod_{k=1}^{MAX\_PRO\_HOP} (1 - \iint_{i\_cirque} P_M)$$

where

P_M=*The probability of area [The right k cirque - the circle whose centre is A and radius is (i+k-1)r] have* $\geq$ *1 beacons*, here A is a point in the i_cirque integral.

(Assume beacon nodes follow Poisson distribution, the probability of x beacons lying in an area of S can be computed by $P(x) = \frac{(D*S)^x}{x!}e^{-D*S}$ [6], where D is the density of beacon nodes, $D = \frac{N_{bea}}{S_{area}}$.)

We calculate the detection probability using matlab to give an intuitive impression. The results are shown in Figure 4(b). Radio range is set to be 15, area is $100\times100m^2$. We set the theoretical hop length u to be 11.7m, because [11] shows that 15 is a critical neighbor number to Hop-Count based localization, and applying formula 1 we get 11.76m. We vary the wormhole length and the beacon node number. And results indicate that the detection probability is high when there are sufficient beacon nodes. Simulation results of detection ratio are presented in Section 6.1.

5.2. Security Analysis

Internal attackers may want to change the hop-count information in beacons' packets. But since they only get the hash results of h_0 and ids, according to the property of hash function, they can't remove preceding nodes in the path to reduce hop-count. Also the successor nodes will check whether the preceding node added its id into the $nodelist$, so it can't forward packets without adding itself into the path. On the other side, the hop-count increment attack, which has the effect of increasing the hop-count field more than 1, turns out to be an ineffective attack. It has little effect on location algorithms, because there are multiple paths to a node which don't pass that malicious node, and other paths of the same length will be selected instead if that malicious node adds the length of the path passing it. Then location results will be almost the same as when malicious nodes don't lie in the networks. This is also confirmed by simulation results, see Section 6.4.

Our wormhole detection mechanism in Section 5.1 has a high probability to detect a wormhole, and it also shows a high probability to detect malicious beacons in Section 6.2. Also, TESLA can enable sensor nodes to authenticate received packets and ensures integrity [13], so node can't forge beacon packets. LMS based location estimation is also resistant to few bad location references (malicious beacons).

6. Simulation Results

We simulate our algorithm in matlab. We randomly place 300 nodes in $100\times100m^2$ area, 30 of them are beacon nodes, radio range is 15, the MAX_PRO_HOP is set to 4. All results are based on 100 independent runs if not state explicitly.

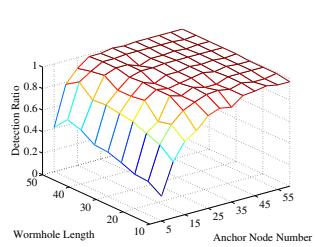
6.1. Detect Ratio of Wormhole Attacks

First we simulate wormhole detection ratio for different beacon numbers and different wormhole lengths. Results are shown in Figure 5(a). We can see that detection ratios are high when there are more than 30 beacons, over 95%.

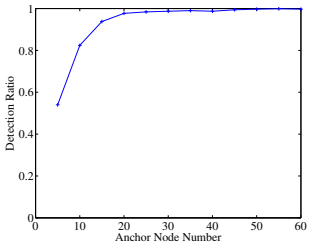
We also simulate in a different manner: we randomly choose two locations in the area to form a wormhole, and then we carry out the detection. Our results based on 1000 runs are show in Figure 5(b). We can see that the detect ratio is already more than 98% when there are 30 beacons.

6.2. Localization With Malicious Beacons

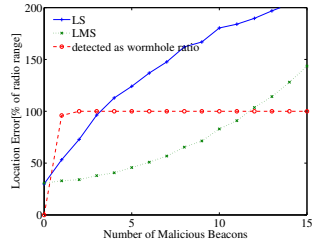
LMS based location estimation with malicious beacons is also simulated. Here if a beacon is malicious, it will report its location as a random selected location from the area. Previous works (e.g., [10]) only study *single-hop* (beacon and node are only 1 hop) case. We can see from Figure 5(c) that LMS performs better than LS when there are malicious beacons. But when the number of malicious beacons is more than 5($\sim$16% of total beacons), LMS location error grows linearly: 12 malicious beacons (40% of total beacons) lead to location error more than 100%, which may be due to the error-accumulation of multi-hop. Fortunately, our wormhole detection mechanism is more sensitive to malicious beacons. When there are merely 1 or 2 malicious beacons, the wormhole detection mechanism will detect them at very high ratio (about 99%) and report them to base station. They tell base station that these beacons are



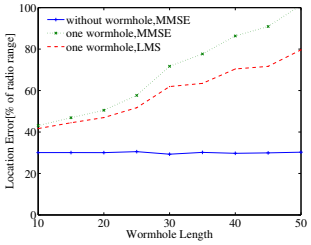
(a) The simulation wormhole detection ratio.



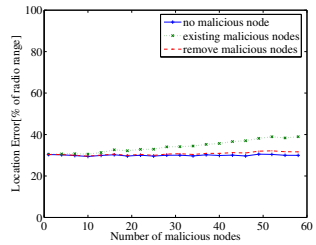
(b) The simulation random length wormhole detection ratio.



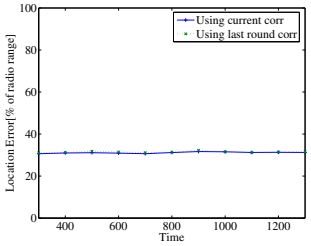
(c) The effect of malicious beacons.



(d) The effect of different wormhole lengths.



(e) The impact of hop-count increment attacks.



(f) The impact of using the corrections of last round.

Figure 5. Simulation results.

malicious beacons or there is a wormhole (must be one of the two cases). The base station can then take actions according to this.

6.3. The Impact of Wormhole Attacks

We simulate the impact of different wormhole lengths to localization, and the resistant ability of LMS to wormhole attacks. Results are shown in Figure 5(d). We can see that LMS based estimation performs a little better than MMSE based estimation, but it still have too high location error, e.g., when there is a 30m length wormhole, location errors of LMS and MMSE are 61% and 71% respectively, comparing to the normal 30%. Results indicate that LMS fails to "filter" these tunneled location references.

6.4. The Impact of Hop-count Increment Attacks

We simulate malicious nodes adding hop-count field (or similar effect, e.g., stuffing $nodelist$ in SHOLOC) of beacon location packets, to evaluate hop-count increment attacks. Here we let malicious nodes add at most 2 hops $max(hop_{current} + 2, MAX\_PRO\_HOP - 1)$, Figure 5(e) shows the simulation results. We can see that even 20% nodes are malicious, they still have little effect on localization results. The location errors of three cases: no malicious nodes, existing malicious nodes, and removing these malicious nodes from network, all are quite approximate. We can conclude that Hop-Count based localization algorithms are inherently robust to hop-count increment attacks, and we only need to prevent the hop-count reduction attacks like we did in SHOLOC.

6.5. The Impact of Using the Corrections of Last Round

We do simulation to decide whether the error introduced by using the corrections of last round is acceptable. When nodes distribute uniformly and move independently, we believe that the random waypoint model is most closest to this situation. We implement that all the nodes (include beacon nodes) run under the random waypoint model, in which nodes in a large area choose some destination randomly, and move there at a random speed chosen from (V_{min}, V_{max}], where V_{min} and V_{max} are the minimum and maximum speed of the nodes respectively. To mitigate the well-known speed decay problem of random waypoint model, we set the V_{min} away from zero, also, we run a swarm up procedure before measurement. In our simulation, V_{min} is set to be 2m/s, and V_{max} is 4m/s. We make a re-localization every 100s, the swarm up procedure endure 200s. Results are shown in Figure 5(f). The location errors of using current correction and using last round correction to calculate location are almost the same.

7. Conclusion

In this paper, we have proposed a secure Hop-Count based localization scheme SHOLOC, which, to the best of our knowledge, is the first secure localization algorithm considering *multihop* situation. We develop a protocol to enable sensor nodes to authenticate received beacon information. We show that only considering filter mechanisms such as LMS can't defeat wormhole attacks and we proposed to employ beacon nodes to detect wormhole. LMS location estimation is also studied in our scheme. Theoretical analysis and simulation results are also given.

For future work, we plan to develop filter algorithms to be used in more complex environments such as concave areas with adversaries.

Acknowledgment

This work is supported in part by the High Technology Research Project of Jiangsu Province of China under Grant No.BG2005029, and in part by the Natural Science Foundation of Jiangsu Province under Grant No.BK2007136.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(2):393C422, 2002.

[2] S. Capkun and J. P. Hubaux. Secure positioning in wireless networks. *IEEE JSAC*, 2006.

[3] W. Du, R. Wang, and P. Ning. An efficient scheme for authenticating public keys in sensor networks. In *Proceedings of MobiHoc*, 2005.

[4] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of INFOCOM*, 2003.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, 1:175–192, 2003.

[6] L. Kleinrock and J. Silvester. Optimum tranmission radii for packet radio networks or why six is a magic number. In *Proceedings of the IEEE NTC*, 1978.

[7] L. Lazos and R. Poovendran. Serloc: Secure range-independent localization for wireless sensor networks. In *ACM WiSe*, 2004.

[8] Z. Li, W. Trappe, Y. Zhang, and B. Nath. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of IPSN*, 2005.

[9] D. Liu and P. Ning. Location-based pairwise key establishments for static sensor networks. In *Proceedings of ACM SASN*, 2003.

[10] D. Liu, P. Ning, and W. K. Du. Attack-resistant location estimation in sensor networks. In *Proceedings of IPSN*, 2005.

[11] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proceedings of IPSN*, 2003.

[12] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *Proceedings of IEEE GLOBECOM*, 2001.

[13] A. Perrig, R. Canetti, D. Tygar, and D. Song. Efficient authentication and signature of multicast streams over lossy channels. In *Proceedings of Oakland*, pages 56–73, May 2000.

[14] P. Rousseeuw and A. Leroy. *Robust regression and outlier detection*. Wiley-Interscience, 2003.

[15] C. Savarese and J. Rabay. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of USENIX*, 2002.

[16] A. Savvides, C.-C. Han, , and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of MobiCom*, Rome,Italy, 2001.

[17] C. Wang and L. Xiao. Locating sensors in concave areas. In *Proceedings of INFOCOM*, 2006.