Handling the Evil Ring Attack on Localization and Routing in Wireless Sensor Networks

Wei Shi, Michel Barbeau, Joaquin Garcia Alfaro, Jean-Pierre Corriveau

Handling the Evil Ring Attack on Localization and Routing in Wireless Sensor Networks

WEI SHI[1⋆], MICHEL BARBEAU[2†], JOAQUIN GARCIA-ALFARO[3‡], JEAN-PIERRE CORRIVEAU[2¶]

[1] *Faculty of Business and Information Technology, University of Ontario Institute Technology, Oshawa, Canada*
[2] *School of Computer Science, Carleton University, Ottawa, Canada*
[3] *Institut Telecom, Telecom Bretagne, LUSSI Department, 02 rue de la Chataigneraie, CS 17607, 35576 Cesson Sevigne, France*

Received March 2011; In final form yyyy 20XX

Compass, face and geographical routing, for ad hoc and wireless sensor networks, rely on nodes knowing their geographic location and locations of other nodes. For location-unaware nodes without self-positiong devices (e.g., GPS), Garcia-Alfaro et al. proposed location determination algorithms leveraging location reports from neighbors.

The *evil ring* is an attack on the location determination algorithms of Garcia-Alfaro et al. When inquired, an attacker returns a fake location sitting on a circle centered at the victim's location and with radius equal to the attacker-victim separation distance. The calculation of the distance between the victim and attacker is not affected. A location-unaware node correctly determines its location. The attack, however, misleads into getting and using wrong locations of neighbors.

We introduce and analyze an evil ring attack detection algorithm. Location-unaware nodes crosscheck consistency of information collected from neighbors. Neighbors perpetrating the evil ring

⋆ email: wei.shi@uoit.ca
† email: barbeau@scs.carleton.ca
‡ email: joaquin.garcia-alfaro@acm.org
¶ email: jeanpier@scs.carleton.ca

attack are uncovered. $O(n)$ messages are sufficient to localize all location-unaware nodes, to detect all liars, and to construct valid neighbor tables.

Simulation results demonstrate that our algorithm, named *Cross Check*, outperforms the Garcia-Alfaro et al. algorithm *Majority-ThreeNeighborSignals*. We compared the percentage of location-unaware nodes that successfully derive valid neighbor tables. Simulations were conducted under equivalent topology conditions, varying the percentage of evil ring attack perpetrators.

Key words: Ad Hoc Network; Wireless Sensor Network; Liar Detection; Localization; Algorithms; Compass Routing; Face Routing; Geographical Routing.

1 INTRODUCTION

For ad hoc and wireless sensor networks (WSNs), there are routing algorithms relying on nodes knowing their own geographic location and locations of others [9, 16, 19]. Such routing algorithms include compass routing, face routing [2], and geographical routing [12]. Nodes equipped with GPS devices can determine their geographic location. There are, however, instances where GPS devices are unavailable or inoperative, possibly due to signal obstruction. Supplementary localization techniques are required.

Localization has been studied by several authors [3, 10, 18]. For the sake of simplicity, from hereafter no measurement error is assumed. Techniques such as *received signal strength* and *time of flight* have been put forth [1]. For secure localization, Capkun and Hubaux used trusted third parties, authenticated distance estimation, authenticated distance bounding, verifiable trilateration, and verifiable time difference of arrival [4]. Their method is tolerant to the distance modification attack, with a large number of adversaries. It is, nevertheless, unable to pinpoint attackers. Hwang et al. [11] proposed a secure localization mechanism detecting existence of attackers, termed phantom nodes. Their approach solely offers stochastic guarantees. Along the same lines, Liu et al. proposed the use of detector nodes in charge of recognizing adversaries [14, 15]. Lazos et al. [13] are the authors of a decentralized method for both secure localization and location verification. The method requires a small number of reference points. Adversaries are limited in their ability to mimic false locations, but they cannot be pinpointed. Sastry et al. proposed eliminating malicious data in a localization process by dropping location references that are inconsistent with references disseminated by a set

of trusted anchors [17]. Delaet et al. presented a deterministic secure positioning algorithm that requires $2n^2$ messages [5]. An assumption of *a priori* knowledge of all node locations is rather constraining.

Garcia-Alfaro et al. proposed location determination algorithms leveraging neighbor location reports and techniques such as *time of arrival*, *time difference* of arrival, and *angle of arrival* [8, 6, 7]. It is assumed that some nodes are *liars* and report false locations. If the number of liars is below a threshold, then location-unaware nodes still determine their correct location by applying majority rules. In this paper, we present an attack on the location determination algorithms of Garcia-Alfaro et al. that misleads into getting and using wrong locations of neighbors. If the attacker sits on a circle centered at the victim's location and of a radius corresponding to the attacker-victim separation distance, then the attack is undetectable. This is due to the fact that the calculation of the distance between the attacker and victim is not modified. If there are enough truth telling nodes, then victims can still determine their correct location. They are, nevertheless, unable to detect the liars. We call this stratagem the *evil ring attack*. It enables attacks against routing protocols that require knowledge of the locations of other nodes for correct operation.

Building on the work of Garcia-Alfaro et al. [6, 7, 8] and Delaet et al. [5], we propose a distributed algorithm for localizing nodes in ad hoc and WSNs in presence of liars and evil ring attack perpetrators. This algorithm enables correct construction of neighbor tables for all nodes, including initially location-unaware nodes. Our algorithm cross checks consistency of the information collected from neighbors. Evil ring attack perpetrators are detected.

In the context of the work of Garcia-Alfaro et al., the *evil ring attack*, is presented in Section 2. Our model and assumptions are defined in Section 3. The evil ring attack detection algorithm is described in Section 4. Correctness and complexity are analyzed in Section 5. Simulation results are discussed in Section 6. We conclude with Section 7.

2 LOCALIZATION ALGORITHM AND ATTACK MODEL

According to the Garcia-Alfaro et al. algorithms, a node U determines its location leveraging neighbor location triplets [6, 7, 8]. Let V_1, V_2, and V_3 denote three neighbors at distances d_1, d_2, and d_3. The location of U is the intersection point of the three circles centered at locations V_1, V_2, and V_3 and radii d_1, d_2, and d_3. We indistinctly refer to a node or its location.

It is assumed that some neighbors lie about their location, but not about their distance. The algorithms are liar tolerant. Algorithm 1 describes the pro-

Algorithm 1 MAJORITY-THREENEIGHBORSIGNALS [8]

1: Node U requests neighbor locations.
2: Every neighbor broadcasts its location.
3: **for all** neighbor triplet (V_1, V_2, V_3) **do**
4: Compute (x, y).
// (x, y) is the intersection point of the three circles
// centered at V_1, V_2, V_3 and with radii d_1, d_2, d_3.
5: **end for**
6: The majority of intersection points determine the location.
// if there is no consensus, then return failure.

cedure *Majority-ThreeNeighbourSignals* executed by every location-unaware node U. Locations of neighbors are requested (Line 1) and returned using broadcast (Line 2). Replies are used to from neighbor triplets and to derive intersection points (Line 3). Several different intersection points may be obtained (Line 4), but the majority determines the location of U (Line 6). There are upper bounds on the number of tolerable liars, otherwise the algorithm fails. As a function of the liar number, Table 1 lists the minimum number of neighbors required to determine a location. Evil ring attack perpetrators

Number of Liars	Min Number of Neighbors
1	7
2	11
3	16
4	21
5	26
10	31
15	74
20	98

TABLE 1: Minimum number of location-aware neighbors required to determine a correct location, as a function of the number of liars [8].

are liars undetectable by the majority-rule of Algorithm 1 (Line 4). Figure 1 pictures the attack. Node V_1 is a liar and is used by U. Part (a) shows that node V_1 can report any location on the (dashed) circle centered at position U and of radius d_1. The distance to node V_1 and intersection point are invariant. Node U is misled into getting an incorrect location for node V_1. The incorrect

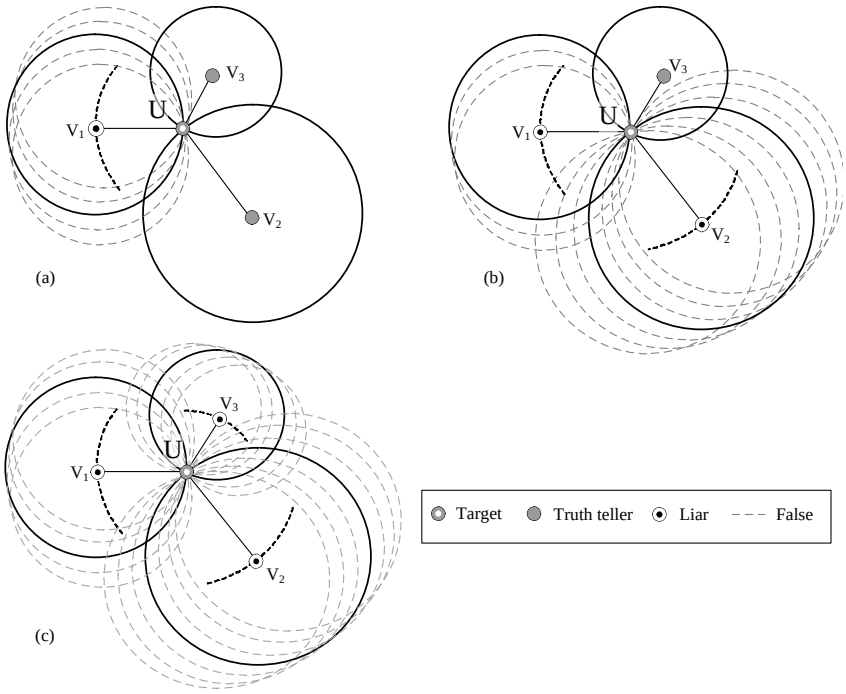


FIGURE 1: The evil ring attack involving one (a), two (b) or three (c) nodes.

location may potentially disrupt the operation of location-based algorithms, such as geographical routing. Parts (b) and (c) show that two or three liars can be involved in a triplet. In the following section, we describe a technique for detecting evil ring attack perpetrators through cross verification of location reports.

3 MODEL AND ASSUMPTIONS

Let $\mathcal{V}$ denote the set of all nodes in a network. $U \in \mathcal{V}$ is a *location-unaware* node. Let $\mathcal{N}$ denote the set of all location-aware nodes in the communication range of U, i.e., the neighbors of U ($U \notin \mathcal{N}$). Let $\mathcal{M}$ denote the set of liars, with $\mathcal{M} \subset \mathcal{N}$. Liar mislead nodes into getting wrong location reports. We assume that liars cannot interfere with the measurement techniques used to determine distances. Beyond the assumptions stated in Ref. [8], we also

assume that no three nodes are collinear. The following theorem formally establishes a weakness of Algorithm 1.

Theorem 1 *The evil ring attack is transparent to the Algorithm* Majority-ThreeNeighborSignals.

Proof: Given the locations of three truthful location-aware nodes V_1, V_2, V_3 and their separation distances d_1, d_2, d_3, a location-unaware node U calculates a location. As illustrated by the solid circles in Figure 2, the location (x, y) is the solution to the following system of equations ($[1]$ and $[2]$ are the standard projection functions) :

$$(V_1[1] - x)^2 + (V_1[2] - y)^2 = d_1^2$$
$$(V_2[1] - x)^2 + (V_2[2] - y)^2 = d_2^2$$
$$(V_3[1] - x)^2 + (V_3[2] - y)^2 = d_3^2$$

If there is at least one liar in the triplet V_1, V_2, V_3, then U either calculates a wrong location or fails to calculate a location. Figures 2 and 3 illustrate two attack models. The perpetrator is V_3. V_3' is the falsely reported location. Algorithm *Majority-ThreeNeighborSignals* applies the following two rules:

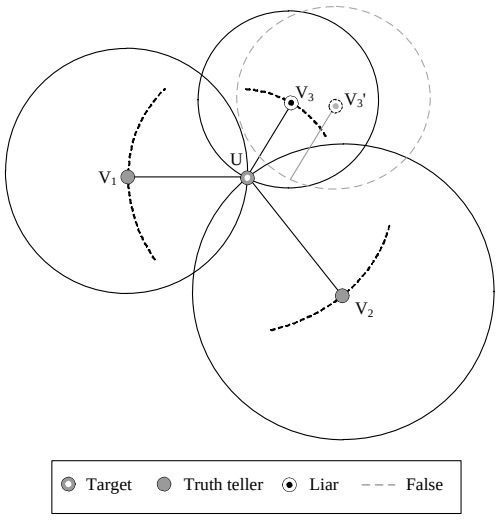


FIGURE 2: Node V_3 reporting a fake location V_3'.

1. If either the location (x, y) is in disagreement with the majority or no location can be calculated, then all three nodes V_1, V_2, V_3 are considered liars.

2. Conversely, if a location (x, y) is in agreement with the majority, then all three nodes V_1, V_2, V_3 are considered truth tellers.

Let us examine the situation pictured in Figure 1 Part (a). Node V_1 is a liar. Fake locations are on a circle, centered at U and of radius d_1, partially represented by a dashed arc. All the dashed gray circles, centered at V_1 and of radius d_1, intersect at U. True location reports and fake location reports result into the calculation of the same location by U. According to the second rule, U concludes that V_1, V_2, V_3 are truth tellers. The fake location reports are transparent to Algorithm *Majority-ThreeNeighborSignals*. Triplets comprising two or three liars of that type are demonstrated in Figure 1 Parts (b) and (c). We conclude that the evil ring attack is transparent to the Algorithm *Majority-ThreeNeighborSignals*. □

In the following section, we describe an algorithm that allows:

1. location-unaware nodes to determine their location,

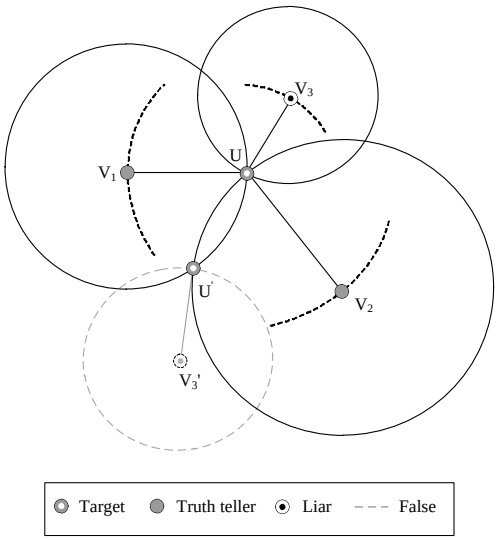


FIGURE 3: Node V_3 reporting a fake location V_3': Node U infers a wrong location U'.

2. location-unaware nodes to detect evil ring attack perpetrators,

3. location-unaware nodes to construct valid neighbor tables, and

4. location-aware nodes to construct neighbor tables including originally location-unaware nodes.

Node U cross checks consistency of information obtained from neighbors in $\mathcal{N}$.

4 EVIL RING ATTACK DETECTION

Algorithm 2 is the detection algorithm executed by a location-unaware node U. Main steps are:

- Request locations (Line 2): Node U sends using broadcast a location request to all the other nodes in the neighborhood.

Algorithm 2 CROSS CHECK for a location-unaware node U

1: **repeat**
2: Request neighbor locations.
3: **for all** triplet of neighbors (V_1, V_2, V_3) **do**
4: Compute the intersection point of the three circles determined by
5: V_1, V_2, V_3 and separation distances d_1, d_2, d_3.
6: **end for**
7: **until** there is a consensus (x, y) determined by a majority of triplets.
8: Accept the location (x, y).
9: **for all** triplet of neighbors (V_1, V_2, V_3) in agreement with the majority **do**
10: Add the locations V_1, V_2, V_3 to the *Cross Check* list.
11: **end for**
12: Broadcast location and *Cross Check* list to neighbors.
13: Wait until reception of two *Cross Check* lists from two different neighbors.
// Check consistency of own and neighbor Cross Check lists
14: **for all** neighbor node **do**
15: **if** node is in all three *Cross Check* lists **then**
16: the node is a truth teller and added to the neighbor table.
17: **else**
18: the node is added to the liar list.
19: **end if**
20: **end for**

- Calculate location (Lines 3 to 8): Using every possible three neighbor combination and their distances, node U calculates a location (x, y) according to the majority rule.
- Build cross check list (Lines 9 to 12): All neighbors in triplets in agreement with the majority are added to the cross check list. The accepted location and cross check list are sent using broadcast to neighbors.
- Liar detection (Lines 13 to 20): Node U waits until it receives two cross lists from two different neighbors. Every node presents with identical location in all three cross check lists is added to the neighbor table. Otherwise, it is added to the list of liars.

Algorithm 3 is implemented by a location-aware node:

- Location broadcast (Line 1): Upon request, location is sent using broadcast.
- Build cross check list (Lines 2 to 8): Node N listens and collects all location reports returned by the neighbors of the requester. Using every possible three neighbor combination and their distances, node N calculates an intersection point. If the calculated intersection point and location of N are equal, then the three neighbors are added to the cross check list. Otherwise, at least one of them is considered lying and the three neighbors are ignored.
- Broadcast of Cross Check list (Line 9): The cross check list is sent to neighbors using broadcast.

Algorithm 3 CROSS CHECK for a location-aware node $N \in \mathcal{N}$

```
1:  Upon reception of a request, broadcast location.
2:  for all triplet of neighbors (V1, V2, V3) do
3:      Compute the intersection point of the three circles determined by
4:      V1, V2, V3 and separation distances d1, d2, d3.
5:      if the intersection point is equal to the location of N then
6:          Add V1, V2, V3 to the Cross Check list.
7:      end if
8:  end for
9:  Broadcast Cross Check list to neighbors.
10: for all received location from a new node U do
11:     Add U to the neighbor table.
12: end for
```

- Discovery of new nodes (Lines 10 to 12): Node N listens and collects information about nodes that have resolved their location. They are added to the neighbor table.

5 CORRECTNESS AND COMPLEXITY ANALYSIS

In this section, correctness of the evil ring attack detection algorithm is demonstrated. The first lemma is about correctness of construction of cross check lists. A cross check list is correctly constructed when all elements are distance truth tellers.

Lemma 2 *Let n be the number of neighbors of a location-unaware node U and m be the number of liars ($m < n$). If*

$$n^3 - 3(2\mathrm{m}+1)n^2 + 2(3\mathrm{m}^2 + 6\mathrm{m} + 1)n - (2\mathrm{m}^3 + 6\mathrm{m}^2 + 4\mathrm{m}) > 0 \quad (1)$$

then the cross check list is correctly constructed by U.

Proof: According to Theorem 1 in Ref. [8], a location-unaware node U implementing Algorithm *Majority-ThreeNeighborSignals* determines a correct location in the presence of m liars when inequality represented by Equation 1 is satisfied. When U accepts a location, Line 8 of Algorithm 2, solely nodes in triplets in agreement with the majority, and hence distance truth tellers, are inserted in the cross check list. □

Lemma 3 *An evil ring attack perpetrator can simultaneously fool at most two nodes.*

Proof: Let M denote an evil ring attack perpetrator. M' is a fake location that M broadcasts upon request. Let's assume that M fools three non-collinear nodes N_1, N_2, and N_3. By definition, M and M' are located on a

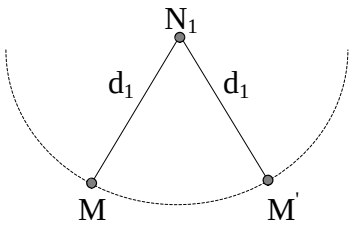


FIGURE 4: Evil ring attacker perpetrator M uses a fake location M' on a circle centered at N_1, with radius d_1.

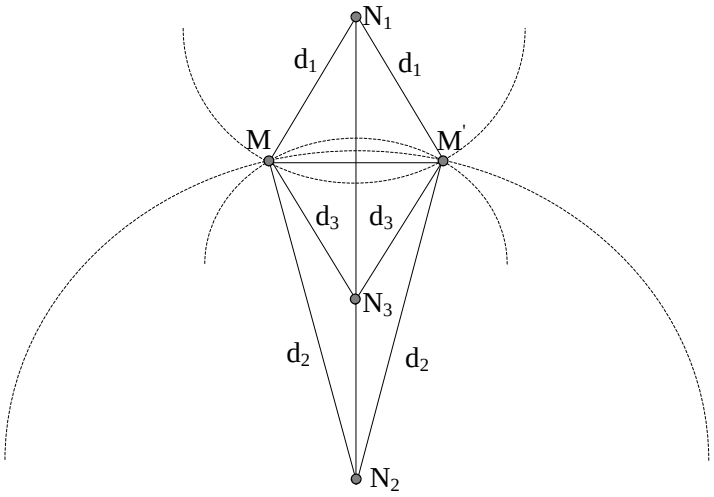


FIGURE 5: The kites N_1MN_2M' and N_1MN_3M'.

circle centered at location N_1 and of radius d_1, see Figure 4. Similarly, M and M' are located on two other circles centered at locations N_2 and N_3, with radii d_2 and d_3. For $i = 1, 2, 3$, the distances N_i to M and N_i to M' are both equal to d_i. The polygon defined by the locations N_1, M, N_2, and M' is a kite, see Figure 5. The polygon defined by the locations N_1, M, N_3, and M' is also a kite. By construction, the diagonals N_1 to N_2 and N_1 to N_3 are both perpendicular bisectors of the diagonal M to M'. The points N_1, N_2, and N_3 are all on the same straight line. This contradicts the assumptions of non-collinearity. Hence, an evil ring attack perpetrator cannot fool simultaneously three nodes or more. □

Lemma 4 *Following its location determination, a node may detect liars and evil ring attack perpetrators with the assistance of two neighbors.*

Proof: Let U be a location-unaware node and M be a liar. M' is a fake location that M broadcasts upon request. According to Algorithm *Majority-ThreeNeighborSignals* [8], if M is not an evil ring attack perpetrator, then U detects M' because the triplets including M' are in disagreement with the majority. As shown in Lemma 2, if M is an evil ring attack perpetrator with respect to U, then M' is listed in the cross check list of U. Lemma 3 has shown that M can fool at most two nodes. Hence, M' is listed in at most

one other cross check list, of some neighbor. The condition of Line 15 of Algorithm 2 is not verified. U puts M in the liar list (Line 18). U adds a node to its neighbor table (Line 16) solely if it is consistently listed in three different cross check lists, i.e. its own and the cross check lists of two other neighbors. □

Theorem 5 *Following the execution of Algorithm Cross Check, a location-unaware node has determined its location and constructed a neighbor table solely including truth tellers. A location-aware node updates its neighbor table with the new nodes in the neighborhood that learned their locations.*

Proof: A location-unaware node can obtain its location using the majority rule described in Ref. [8]. According to Lemma 3, the node can detect the evil ring attack perpetrators with assistance from two neighbors. According to Lines 14 to 20 of Algorithm 2, a node is added to the neighbor table solely if it is a truth teller. According to Lines 10 and 12 of Algorithm 3, a location-aware node adds to the neighbor table nodes that resolved and advertised their location. □

Theorem 6 *The message complexity of Algorithm Cross Check is $O(n)$, where n is the number of network nodes.*

Proof: Let's assume that k location-unaware nodes send using broadcast a total of k requests to the other $n-k$ nodes. The $n-k$ location-aware nodes send using broadcast a total of $n-k$ replies. All n nodes send using broadcast their cross check list. Thus the total number of messages is $k+n-k+n$ or $O(n)$. □

6 SIMULATION RESULTS

We compared experimentally the algorithms Majority-ThreeNeighborSignals and Cross Check. In presence of evil ring attack perpetrators and under equivalent topology conditions, the percentages of location-unaware nodes with valid neighbor tables are compared.

The simulation model consists of m sensors randomly and uniformly distributed over an unit square. The communication range of each sensor is a circle centered at its location with radius $r = \sqrt{\frac{\ln m + l \ln\ln m + \ln(l!) + c}{m\pi}}$, as proposed in Ref. [3]. The integer parameter $l \geq 0$ determines the network connectivity. A network is $l+1$-connected if it remains connected when up to l nodes are deleted. The real number constant c determines the probability that

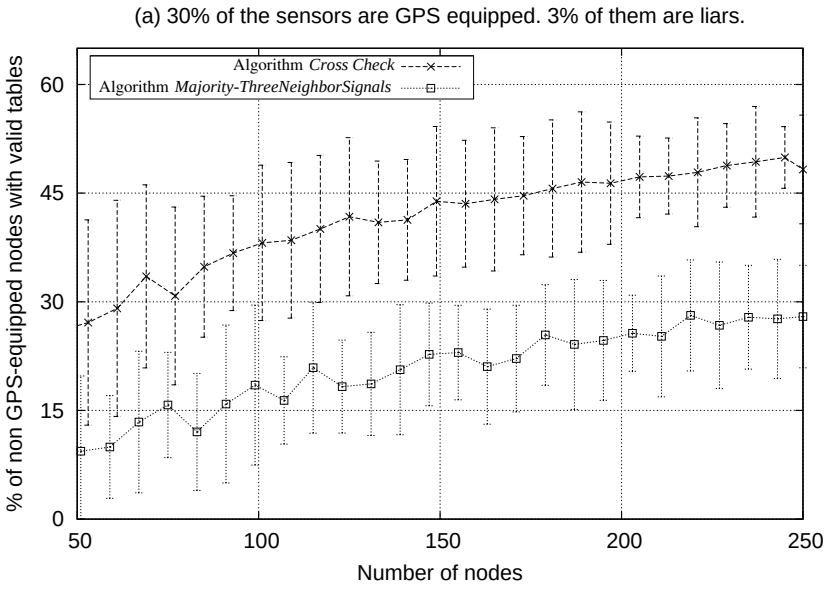
the network is $l + 1$-connected (cf. [3] and citations thereof). We assume that both l and c are equal to one. The simulation model has been implemented in Java. Fake locations of liars are selected at random. Fake locations of evil ring attack perpetrators are selected also at random, but on circles determined by the locations of the victims and distances to the true locations.

We executed four sets of simulations. Every simulation set consists of 50 to 250-node WSNs. An average of 30% of the nodes are GPS equipped, i.e., location-aware. The non GPS-equipped nodes are initially location unaware. If at least one entry is incorrect, then the neighbor table of a non GPS-equipped node is considered invalid. For each simulation, we compute the percentage of non GPS-equipped nodes that build valid neighbor tables. Figures 6 and 7 plot the results. The indicated 95% confidence intervals have been obtained with 100 simulations.

Results are presented in increasing percentage of evil ring attack perpetrators, 3%, 7%, 10%, and 15% of the total numbers of sensors. The x-axis represents the total numbers of nodes in the simulated WSNs. The y-axis represents the percentages of non GPS-equipped nodes building valid neighbor tables. Algorithm Cross Check always leads to networks with higher percentages of non GPS-equipped nodes with valid neighbor tables. Figure 6 Part (a) shows that, when 10% of the GPS-equipped nodes are evil ring attack perpetrators, with Algorithm Cross Check about 30% more nodes build valid neighbor tables in the 50 to 150-node WSNs; and almost 40% more in the 150 to 250-node WSNs. Part (b) shows that, when 25% of the GPS-equipped nodes are evil ring attack perpetrators, 40% more nodes construct valid neighbor tables in the 50 to 150-node WSNs; and about 50% more in the 150 to 250-node WSNs. Finally, Figure 7 Parts (c) and (d) show that almost the same proportions hold in scenarios where 35% or 50% of the GPS-equipped nodes are evil ring attack perpetrators.

7 CONCLUSION

We have presented the evil ring atack on the localization algorithms of Garcia-Alfaro et al. [6, 7, 8]. The attack enables other attacks on routing protocols requiring the locations of other nodes. We have formally demonstrated the execution of the attack. We have also proposed an algorithm that detects the evil ring attack. The correctness of the algorithm has been demonstrated. Its complexity has been analyzed. Simulation results support the claim that evil attack perpetrators are detected and neighbor tables are constructed solely with valid entries.



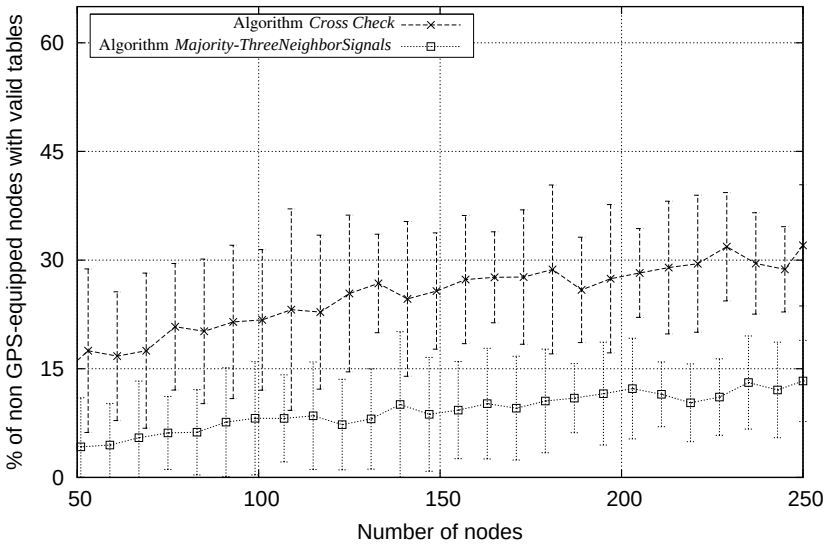
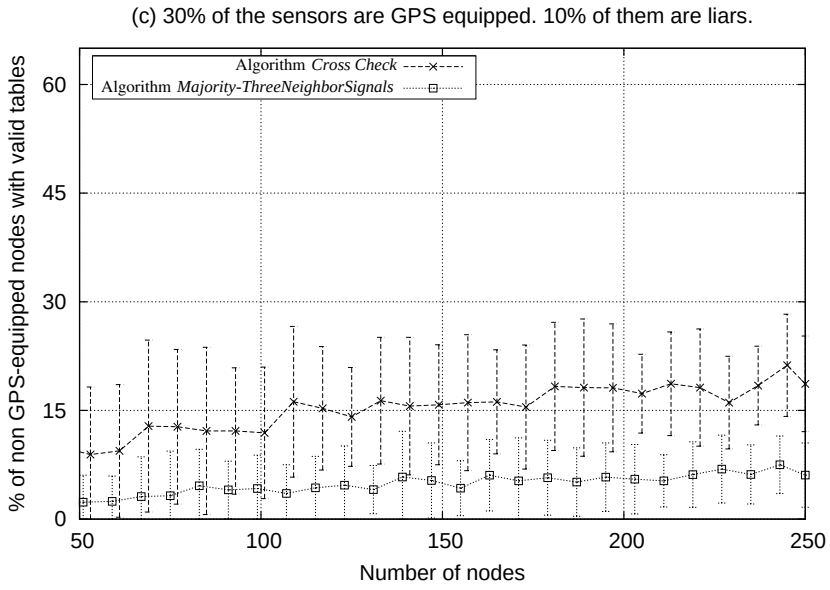


FIGURE 6: Simulation results. (a) 30% of the sensors are GPS-equipped nodes. 10% of them are evil ring attack perpetrators. The curves represent the percentage of non GPS-equipped nodes that, after executing Algorithm *Cross Check* or Algorithm *Majority-ThreeNeighborSignals*, succeed at computing valid neighbor tables. (b) 25% of the GPS-equipped nodes are evil ring attack perpetrators.



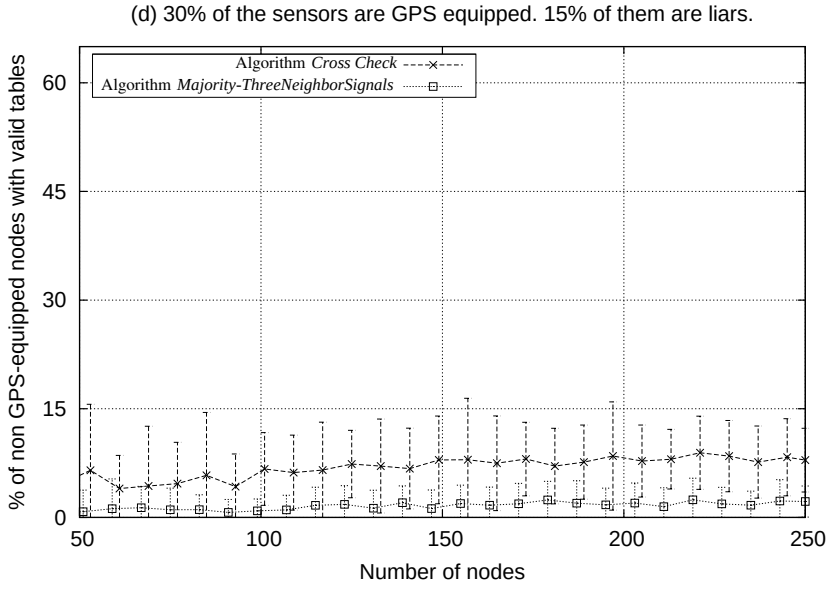


FIGURE 7: Simulation results. (c) 35% of the GPS-equipped nodes are evil ring attack perpetrators. (d) 50% of the GPS-equipped nodes are evil ring attack perpetrators.

8 ACKNOWLEDGMENTS

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Spanish Ministry of Science (TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER-INGENIO 2010 CSD2007-00004 A-RES grants) is graciously acknowledged.

REFERENCES

[1] P. Bahl, V.N. Padmanabhan, and A. Balachandran. (2000). Enhancements to the RADAR user location and tracking system. Technical Report MSR-TR-2000-12, Microsoft Research. 13 pages.

[2] M. Barbeau and E. Kranakis. (2007). *Principles of Ad Hoc Networking*. Wiley and Sons Ltd.

[3] M. Barbeau, E. Kranakis, D. Krizanc, and P. Morin. (2004). Improving distance based geographic location techniques in sensor networks. In *The 3^{rd} International Conference on AD HOC Networks & Wireless (ADHOC-NOW'04), volume 3158 of Lecture Notes in Computer Science*, pages 197–210.

[4] S. Capkun and J. P. Hubaux. (2006). Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications: Special Issue on Security in Wireless Ad Hoc Networks*, 24(2):221–232.

[5] S. Delaet, P. Mandal, M. Rokicki, and S. Tixeuil. (2008). Deterministic secure positioning in wireless sensor networks. In *DCOSS 2008: IEEE International Conference on Distributed Computing in Sensor Networks, Lecture Notes in Computer Science,Volume 5067*, pages 469–477, Springer Berlin/Heidelberg.

[6] J. Garcia-Alfaro, M. Barbeau, and E. Kranakis. (2009). Secure localization of nodes in wireless sensor networks with limited number of truth tellers. In *The 7^{th} Annual Communication Networks and Services Research (CNSR) Conference, IEEE Communications Society*, pages 86–93, Moncton, New Brunswick, Canada.

[7] J. Garcia-Alfaro, M. Barbeau, and E. Kranakis. (March 2010). Positioning of wireless sensor nodes in the presence of liars. Technical Report TR-10-04, School of Computer Science, Carleton University.

[8] J. Garcia-Alfaro, M. Barbeau, and E. Kranakis. (2011). Secure geolocalization of wireless sensor nodes in the presence of misbehaving anchor nodes. *Annals of Telecommunications*, 66(9–10):535–552.

[9] S. Giordano, I. Stojmenovic, and L. Blazevic. (2003). Position based routing algorithms for ad hoc networks: A taxonomy. *Ad Hoc Wireless Networking*, pages 103–136.

[10] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher. (2003). Range-free localization schemes for large scale sensor networks. In *9th ACM Annual International Conference on Mobile Computing and Networking*, pages 81–95.

[11] J. Hwang, T. He, and Y. Kim. (2008). Secure localization with phantom node detection. *Ad Hoc Networks*, 6(7):1031–1050.

[12] B. Karp and H.T. Kung. (2000). GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom 2000: Proceedings of the 6^{th} annual international conference on Mobile computing and networking*, pages 243–254, Boston, Massachusetts, United States.

[13] L. Lazos, R. Poovendran, and S. Capkun. (2005). ROPE: Robust position estimation in wireless sensor networks. In *The 4^{th} Intl symposium on Information processing in sensor networks*, pages 43–50.

[14] D. Liu, P. Ning, and W. Du. (2005). Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *25th IEEE International Conference on Distributed Computing Systems*, pages 609–619.

[15] D. Liu, P. Ning, A. Liu, C. Wang, and W.K. Du. (July 2008). Attack-resistant location estimation in wireless sensor networks. *Transactions on Information and System Security*, 11(4):1–39.

[16] M. Mauve, J. Widmer, and H. Hartenstein. (2001). A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30–39.

[17] N. Sastry, U. Shankar, and D. Wagner. (2003). Secure verification of location claims. In *2nd ACM Workshop on Wireless Security*, pages 1–10.

[18] A. Savvides, C. Han, and M. Strivastava. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. In *7th ACM annual international conference on Mobile computing and networking*, pages 166–179.

[19] I. Stojmenovic. (2002). Position-based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134.