# How Secure are Secure Localization Protocols in WSNs?

Chérifa Boucetta, Mohamed Ali Kaafar, and Marine Minier

INRIA, France

**Abstract.** Remote monitoring and gathering information are the main objectives behind deploying Wireless Sensor Networks (WSNs). Besides WSN issues due to communication and computation restricted resources (low energy, limited memory computational speed and bandwidth), securing sensor networks is one of the major challenges these networks have to face. In particular, the security of sensors localization is a fundamental building block for many applications such as efficient routing.

In this paper, we introduce a new threat model that combines classical Wormhole attacks (i.e. an attacker receives packets at one location in the network, tunnels and replays them at another remote location using a powerful transceiver as an out of band channel) with false neighborhood topology information sent by the wormhole endpoints themselves or by some colluding compromised nodes. We show using intensive simulations how this clever attacker that would exploit the neighborhood topology information can easily defeat two representative secure localization schemes. We also present some possible countermeasures and the first corresponding results.

## 1 Introduction

Wireless Sensor Networks (WSNs) are composed of a large number of low-cost, low-power and multi-functional sensor nodes that communicate at short distances through wireless links. The sensor nodes cooperate together to collect, transmit and forward data to particular points called base stations. Most of the time, they are deployed in an open and uncontrolled environment where attackers may be present.

Due to the lack of infrastructure and to the ease of physical layer exploits, many security threats could be considered in the WSNs context. Many of those threats target the accuracy of localization information. In this case, the aim of the attacker is to construct a false topology to divert the traffic to her own advantage and then to launch traffic analysis, selective-forwarding attacks or Denial of Service (DoS) attacks, etc. Sybil attacks [1], Wormhole attacks [2], false neighbors information [2] are the major security threats for the security of localization in WSNs.

In this paper, we consider both Wormhole attacks and false neighbors information:

G. Parr and P. Morrow (Eds.): S-Cube 2010, LNICST 57, pp. 164–178, 2011.

- Wormhole attack is particularly harmful against routing in sensor networks where an attacker installs a dedicated connection between two distant points (called wormhole endpoints) by a variety of means (e.g. a network cable, a long-range wireless transmission in a different band, etc.). Then, she overhears packets at one extremity, tunnels them through the wormhole link to another point in the network. The second extremity broadcasts the packets among its neighborhood nodes. The created tunnel creates the illusion of proximity between the two endpoints.
- In a false neighbors information attack, an attacker lies about the list of her direct neighbors (i.e. sensor nodes within range). She could include nodes that are close to the base stations (Sinkhole attacks) creating a false topology in the network.

Those two particular attacks have almost the same objectives in the network: to control a portion of the network by creating a false topology. Such topology's correctness is important from several applications' perspective, efficient routing being one of the most notable. Extensive previous researches have proposed several ways to secure localization in WSNs. So far, the paradigm of secure localization protocols is that once securing one of these two attacks, the second one is by design canceled. For instance, if one is sure that there is no wormhole attack in the network, he might have a good confidence (running a simple distance bounding test) that the neighborhood information is safe. Based on this paradigm two classes of protocols have emerged. On the one hand, some solutions are dedicated to defend against false neighbors information [4,7,9,10]. Most of them, if not all, claim to defend against wormhole attacks. They are essentially based on dedicated hardware or on statistical and geometrical tests coupled with local neighborhood information. On the other hand, several other solutions aim to specifically defend against wormhole attacks [3,5,6,8], often using the same principles than protocols securing neighborhood information.

**Contributions.** Although the neighbors discovery process might be secured to compute accurate neighbors lists, current research literature has ignored so far the existence of both threats, i.e. wormhole endpoints sending also biased lists of neighbors, and so trying to mislead either wormhole detection mechanisms or "secure" neighbors discovery processes.

In this paper, we study how potent this danger is, for two representative approaches: the first aims in defending against wormhole attacks through local neighborhood information [6], and the second approach [7] proposes a neighbor-verification protocol designed to secure localization in WSNs. We first introduce a simple, yet novel threat model where the wormhole attackers are not only relaying packets through physical tunnels, but also lie consistently according to such a tunnel. We then study the impact of co-existence of both threats in an WSN. We identify different scenarios to perform these attacks and we demonstrate that it is easy to hide the wormhole attack.

We also show that these clever wormhole attackers would lead to high false positive ratios, that prevent honest, uncompromised sensors, to properly function. We also study how conspiracy can be achieved, and how much it could

affect the security protocols. The "effectiveness" of these attacks on the studied approaches are demonstrated through extensive simulations. Finally we provide countermeasures that can be implemented in today's solutions so as to prevent our attacks.

**Assumptions and Threat Model.** We assume that attackers have access to the same data as legitimate nodes. An adversary is able to send misleading information when probed, or send manipulated information after receiving a request or affect some metrics observed by chosen targets. An adversary can be a wormhole endpoint, or simply a node lying about its neighborhood or both. Based on these assumptions, we identify four types of attacks scenarios that we describe in section 4.

**Paper Organization.** The rest of the paper is organized as follows. Section 2 provides a brief overview of secure localization protocols. In section 3, we describe in more details the workings of the two approaches, chosen for this study. We identify and classify our attacks in section 4. We demonstrate and study the effects of theses attacks, through extensive simulations, in section 5. In section 6, we discuss ways to prevent our attacks, and provide easy to implement countermeasures. Finally, section 7 concludes this paper.

## 2 Background

In this section, we give a brief survey of recently proposed approaches for securing localization in WSNs. Most of these approaches aim to defend against wormhole attacks. They can be classified in two categories.

### 2.1 Bounding Distance-Based Approaches

Distance bounding protocols are used to verify that a node, say $u$, being at a distance $d_{uv}$ from a verifier node $v$, is not providing a shorter distance, say $d'_{uv}$, than what it actually is (i.e. check if $d'_{uv} < d_{uv}$) [11].

In [3] Hu et al. use packet leashes to detect wormhole attacks. Packet leashes contain geographical or temporal information to bound the distance or the lifetime of a end-to-end transmitted packet to restrict its travel to the destination. The sender includes a timestamp or a localization information in the message and the receiver checks that the packets-receiving is in "legal" time or in legal distance. However, this method requires either precise location information obtained via an out-of-band mechanism like GPS or needs loosely clock synchronization between the nodes.

In [5], the authors propose an authenticated distance bounding technique called MAD protocol. In this protocol, each node computes distance using timing properties to verify whether a node is a true neighbor. MAD protocol needs a special hardware radio to switch very quickly between both send and receive modes. The authors of [12,13] proposed an anchor-based scheme. Each node estimates the distance to anchor nodes by using a hop-counting technique. The

disadvantage of this solution is that it mainly relies on anchor nodes which needs, besides trusting these anchor nodes, a preliminary manual setup and an a priori deployment knowledge.

### 2.2 Graph Theoretic and Geometry-Based Approaches

Hu and Evans [4] propose the use of special hardware called directional antennas, to secure localizations in WSNs. Each node executes a neighbors discovery process to construct neighbor list using directional antennas in each direction. Only when the directions of both pairs match, the neighboring relationship is confirmed. However, the use of directional antennas limits the use of this protocol as sensors would be costly.

In [14] authors presented a centralized wormhole detection technique, which adds a virtual network layout using multi-dimensional scaling (MDS) technique. The proposed algorithm tries to establish a virtual position for every node. It respects the constraints induced by the connectivity and the distance estimation data. This technique is unfortunately significantly sensitive to distance estimation errors.

Additional work by [15] uses dedicated nodes called "guard nodes" in a theoretic framework to prevent wormhole attacks. Guard nodes know their actual locations and have higher transmit power. The usage of such special-purpose-guard nodes makes this approach inadequate to WSNs, since compromising guard nodes would lead to the failure of the security protocol.

The scheme proposed in [8] detects wormhole attacks using connectivity information. Each node is represented by a disk, the radius of which being the range of the node itself. The detection algorithm looks for forbidden geometrical substructures in the connectivity graphs that would appear under a wormhole attack. More precisely, a wormhole link will create false disk intersections and thus could be detected. The precise form of these structures depends on the connectivity pattern.

## 3 Approaches for Detecting Wormhole Attacks in Our Study

In our work, we choose to concentrate on two promising and representative approaches: the first scheme uses graph theory and is purely based on local neighborhood information. The second scheme relies on node's distance estimation and simple geometric tests. Our choice is motivated by the fact that each of these approaches represents a class of protocols to secure localization. Both protocols achieve a high level of security against wormhole attacks. Finally, the two approaches design are totally different as one is designed to detect wormhole attacks and hence the neighborhood information would be safe, whereas the second approach adopts a methodology consisting in securing first the neighborhood discovery process and then detecting wormhole attacks.

### 3.1 LNI: Detecting Wormhole Attacks Using Local Neighborhood Information

In [6], authors presented a scheme for detecting wormhole attacks in wireless sensor networks using *Local Neighborhood Information* (LNI). They assume that the network is dense and static and links are bidirectional. The main idea of this algorithm is that every sensor node can compute a so-called connectivity degree based on a neighborhood verification protocol. The connectivity degrees of the node's neighbors and the node itself are then used to verify the presence or not of a wormhole in the network.

More precisely, the authors of [6] base their wormhole detection algorithm on the following principle: a wormhole endpoint $X_1$ will pretend to have neighbors at one or two hops (i.e. the other wormhole endpoint $X_2$ and its neighbors) that the actual neighbors of $X_1$ cannot see.

To calculate the connectivity degree, LNI uses the edge clustering coefficient (ECC), which was defined in [16], as the number of triangles to which a given edge belongs, divided by the number of triangles that might potentially include it, given the degrees of the adjacent nodes.

This approach could be easily generalized to other "cyclic geometrical structures" such as squares or pentagons. Thus, LNI defines the edge-clustering coefficient $C^g_{i,j}$ of order $g$ where $g$ is the number of points of the cyclic structure. Then, $C^g_{i,j} = \frac{CS^g_{i,j}}{S^g_{i,j}}$ where $CS^g_{i,j}$ is the number of "cyclic structures" of order $g$ the edge $(i, j)$ belongs to, while $S^g_{i,j}$ is the number of all possible cyclic structures of order $g$ that can be built given the degrees of the nodes.

The authors modified the coefficient $CS^g_{i,j}$ to exclude a particular node $x$: they introduce the coefficient $CS^g_{i,j\setminus x}$ which is the number of cyclic structures that exclude $x$. Indeed, a node $j$ could be detected as a wormhole endpoint if one of its neighbors $i$ checks that $\forall x \in V1(j)$, $CS^{g=3}_{i,x\setminus j} = 0$ and $CS^{g=4}_{i,x\setminus j} = 0$, where $V1(j)$ is the set of the 1-hop neighbors of $j$. This means that $i$ can reach the neighbors of $j$ only via $j$ which is really rare in a dense network. The authors limit their study to the case where $g$ =3 or 4 leading that each node exchanges with its neighbors its 2-hops neighborhood.

The complete LNI algorithm works in three steps: Neighborhood discovery, $CS$ computation and isolation phase. This algorithm is decentralized, distributed and runs locally at each node of the network.

In the first step, network nodes exchange $HELLO$ messages to determine their 1-hop (resp. 2-hop) neighborhood list ($V1$) (resp. ($V2$)). Once these lists are constructed, each node $i$ executes the following steps for all of its 1-hop neighbors.

Then, the node $i$ computes $CS^3_{i,k\setminus j}$ for every $k$ in $V1(j)$. If the value is null then $i$ computes $CS^4_{i,k\setminus j}$. If the second value is also null, $i$ declares $j$ as a malicious node, broadcasts an alert message containing $j$'s identity and inserts it in its, so-called red list.

Finally, each node that hears such a message adds $j$ to its red list or increments the corresponding counter of node $j$. When the $j$'s counter reaches a given

threshold, node sends back alert messages to all its direct neighbors to isolate the node $j$ from the network.

Simulation results show that the probability to detect a single wormhole is effectively high and that the number of false positives is relatively low as soon as the degree is sufficiently large. However, as we will show in the next sections, simply sending biased lists of neighbors may affect these results.

### 3.2 SNV: Secure Neighbor Verification Protocol

This second approach [7] proposes a Secure Neighbor Verification (SNV) protocol and it relies on node's distance estimation and simple geometric tests. First, SNV assumes that each sensor node is equipped with two network interfaces: a radio frequency (RF) and a sound interface (US) and that all network nodes can perform cryptographic operations with a symmetric key to secure the exchange of messages. Again, the network is assumed dense and distance between connected nodes to match to a polygon on a plane. The adversary is assumed to control a relay network composed of a set of relay nodes. These are fully connected by out-of-band relay links.

The proposed scheme contains three phases. In the first step, called ranging, every node uses neighbor discovery process and attempts to build its neighbor list and to calculate the distance to each neighbor using ultrasound ranging [7]. In the second step, called Neighbor Table Exchanges, every node shares with each of its neighbors, in an authenticated way, its established neighbors table including the distances calculated in the ranging phase, building then a 2-hop neighbor table. Finally, in the link verification phase each sensor runs locally three consistency tests on the 2-hop neighbor table. Consequently, an attack can be detected and links can be either discarded or confirmed. The three consistency tests are the following:

- Link symmetry test: In the 2-hop neighbors table of $i$, denoted $NT2_i$, any link $(u, v)$ for which the distance measured by $u$ is different from the distance measured by $v$ is discarded. Also, links with only one measurement are deleted. An attack is detected if a fraction of more than $\Theta_{sym}$ such links exists, where $\Theta_{sym}$ represents the acceptable fraction of links with asymmetric length [7].
- Maximum range test: node $i$ deletes links in $NT2_i$ which are longer than the range $R$. An attack is detected if a fraction of more than $\Theta_{range}$ links is discarded. $\Theta_{range}$ represents the acceptable fraction of links reported to be longer than $R$, due to the distance measurement error [7].
- Quadrilateral test: In this test a node $i$ looks at every 4-clique in its 2-hop neighbor table $NT2_i$. In graph theory, a 4-clique is a clique where $i$ knows the lengths. In an undirected graph, a clique is a subset of its vertices such that every two vertices in the subset are connected by an edge. If any of the 4-clique is not a quadrilateral within error tolerance, an attack is detected. Moreover, if a link is part of a convex quadrilateral, it is declared verified.

Simulation results show that the proposed scheme is efficient to detect wormhole attacks in the network. Authors do also show that the LNI algorithm can be successfully used to detect $k$-end relay attacks (more than 2 relay nodes in the wormhole attack). Although we did not consider these aggressive wormhole attackers, we will show in section 5 that even simple 1-end wormhole attacks can be successfully hidden by providing biased lists of neighbors that are consistent with the created wormhole.

## 4 Simple Attacks

In this section, we present the adversary model. Our purpose is to study the impact of attacks consisting on sending biased (falsified) list of neighbors and the effectiveness of the proposed approaches while facing such adversary model. We enumerate four possible attacks' scenarios:

1. As in previous research, wormhole attackers are randomly established between two nodes with a distance greater than the range of sensors. Attackers may drop or replay the data messages.
2. Nodes may lie about their neighborhood. Rather than physically performing wormhole attacks (by means of special hardware), these malicious nodes include a few distant nodes which are not actually in their 1-hop neighborhood, but are neighbors of a similar malicious node considered as the second endpoint of the attack. Both endpoints only lie about their neighborhood when asked, providing the total set, or just a subset, of each others' neighbors. Figure 1 illustrates an example of such an attack scenario. It is worth noticing that in this case, we do no more consider wormhole attacks in the network, but only neighborhood liars. In figure 1a, nodes $x_1$ and $x_2$ represent the wormhole endpoints as in the first scenario. In figure 1b, dashed links represent new links created when node $x_1$ lies about its neighborhood. In essence, node $x_1$ adds a small subset of $x_2$'s neighbors in its own one hop-neighbors lists. In this case, $x_1$ adds only two nodes, namely $j$ and $k$.
3. An attacker may combine both wormhole and neighborhood lie attacks with the hope to hide detection. In such a case, a wormhole endpoint, in addition to establishing a physical tunnel to the second endpoint, also sends falsified list of neighbors to escape detection tests. That is to say, scenario 3 is a combination of both scenarios 1 and 2.
4. The attacks described above can either be carried out by malicious nodes in an independent manner or as a conspiracy created by colluding nodes. Collusion is likely in a scenario where attack propagation happens through the now well tested means of worms. Both wormhole attackers and neighbors are then sending biased neighbors lists, claiming they are each actually neighbors. Neighborhood information is falsified according to the wormhole endpoints. In figure1c, nodes $c$ and $b$, if they are compromised may then claim that in addition to $x_1$, nodes $j$, $l$ and $x_2$ are among their neighbors. Node $x_1$ adds also nodes $j$ and $k$ to its one hop neighbor list.

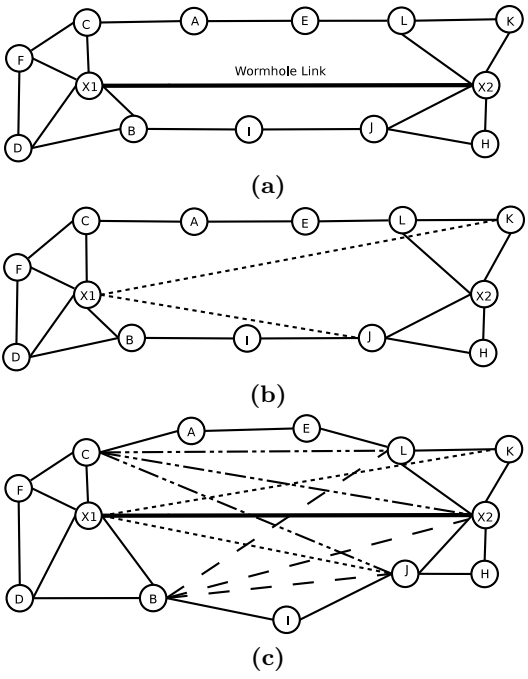


**Fig. 1.** Adversary model: (a) $x_1$ and $x_2$ represent the wormhole endpoints (b) Nodes lying about their neighborhood: $x_1$ adds a small subset of $x_2$'s neighbors in its own one hop-neighbors lists (c) colluding nodes: nodes $x_1$, $c$ and $b$ claim that nodes $k$, $j$, $l$ and $x_2$ are among their neighbors.

In summary, when a wormhole endpoint lies about its neighbors, it may include the other wormhole extremity, and probably few other nearby nodes to hide behind. To study the impact of the different attacks, we mixed both wormhole attacks and neighborhood lies to allow the one or the other to be more successful.

## 5 Experimental Results

We performed extensive simulations to demonstrate the effectiveness of the attacks in misleading the studied secure localizations protocols. In particular, we evaluate the probability of successful detection and mis-detection of attackers in the network.

### 5.1 Simulation Set Up

We performed our simulations on the WSNet simulator [17]. The network consists in 400 static nodes distributed over a square field of 500×500m. The transmission range of each node is 50m. In order to run our experiments on dense sensor environments, as assumed by the studied approaches, the average of local node degree is set to 12. The duration of each experiment is 250 seconds.

Wormhole attacks are randomly established between two nodes with a distance greater than 4 hops. Because of the use of a stationary network, malicious nodes are defined at the time of network establishment. The percentage of malicious nodes varies from 0% (malicious-free system) to 20%. As mentioned in section 4, a malicious node can be a wormhole endpoint, a node lying about its neighborhood or both.

### 5.2 Performance Metrics

To characterize the performance of detection of the wormhole attack, we use the classical false/true positives/negatives indicators. Specifically, a *negative* is a normal node which should therefore be accepted by the test. A *positive* is a malicious node (e.g a wormhole endpoint) which should therefore be rejected by the test and detected as such. The number of negatives (resp. positives) in the population comprising all the network nodes is $PN$ (resp. $PP$).

*A false negative* is a malicious node that has been wrongly classified by the test as negative, and has therefore been wrongly completed. *A false positive* is a normal node that has been wrongly rejected by the test and therefore wrongly aborted. *True positives (resp. true negatives)* are positives (resp. negatives) that have been correctly reported by the test and therefore have been rightly aborted (resp. completed). The number of false negatives (resp. false positives, true negatives and true positives) reported by the test is $TFN$ (resp. $TFP$ , $TTN$ and $TTP$).

We use the notion of *false positive rate* ($FPR$) which is the proportion of all the normal nodes that have been wrongly reported as positive by the test, so $FPR = TFP/PN$. Similarly, the *true positive rate* ($TPR$) is the proportion of malicious nodes that have been rightly reported as malicious by the test, and we have $TPR = TTP/PP$.

### 5.3 Simulations Results

**LNI, Detecting Wormhole Attacks Using Local Neighborhood Information.** Figure 2 shows the variation of the true positive rates as a function of the percentage of malicious nodes. It is clear that the accuracy of detection decreases as the average of malicious nodes increases. It is also interesting to note that when wormhole attackers are lying about their neighborhood, they are detected more often than when they only perform wormhole attacks. In other words, the probability of detecting wormhole attackers lying about their neighbors is higher. This is because, if every wormhole endpoint includes a few neighbors of the other wormhole ends as its own neighbors without consistently checking other endpoints claims and more importantly without its claim being confirmed by its neighbors, alert messages increase inside the network. This simply increases the detection rate. However, we notice that when wormhole attacks are supported by a very few other colluding nodes, and these nodes are all lying about their neighborhood, the detection rate decreases significantly. In particular, when the colluding nodes lie very consistently, i.e. by simply including all

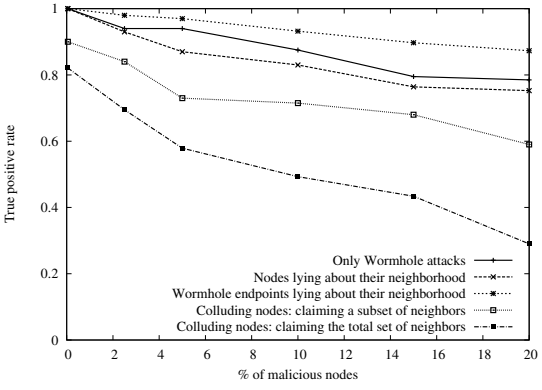


**Fig. 2.** LNI: True Positive Rate *vs* attacks intensity

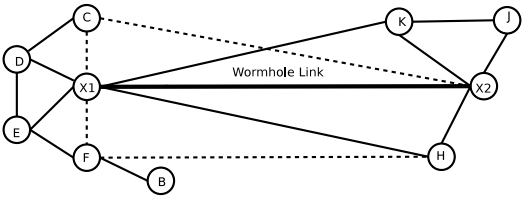


**Fig. 3.** Example of wormhole ends lying about their neighborhood. $X_1$-$X_2$ is the wormhole link

false neighbors as their actual neighbors, the detection rate achieves less than 50%, when the network comprises only 10% of attackers.

We illustrate the previous fact in Figure 3: when a wormhole endpoint lies about its neighbors, the number of nodes that detect it as a malicious node increases. Indeed, when wormhole endpoints, $X_1$ and $X_2$ are lying, and in particular when $X_1$ adds $K$ and $H$ in its neighbors list, $C$ computes $CS^3_{C,X_2 \setminus X_1}$, $CS^4_{C,X_2 \setminus X_1}$, $CS^3_{C,K \setminus X_1}$, $CS^4_{C,K \setminus X_1}$, $CS^3_{C,H \setminus X_1}$ and $CS^4_{C,H \setminus X_1}$. All these values are null, and hence the LNI approach suggests $C$ to declare for instance $X_1$ as a suspicious node 3 times, due to the additional links $(X_1K)$ and $(X_1H)$. This behavior however may have an impact on the number of false positives reported by the test.

Figure 4 shows the variation of false positive rates as function of the percentage of malicious nodes. As expected, the higher the number of nodes lying about their neighborhoods is, the greater the number of false positives is (i.e. the more the test incorrectly classify honest nodes as behaving abnormally). In particular, when the neighbors of the colluding nodes lie consistently, i.e. in concordance with the claims of the wormhole endpoints themselves (scenario (4) described in section 4), the False positive rate increases faster. This is illustrated in figure 4 by the curve "colluding nodes" indicating a high false positive rate of more than

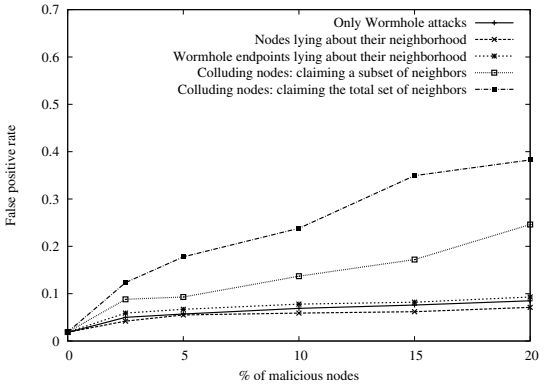


**Fig. 4.** LNI: False Positive Rate *vs* attacks intensity

13% (resp. 20%) when including a subset (resp. the total set) of neighbors, the system being under a low intensity of attack (10% of malicious nodes).

In the case of simple wormhole attackers or wormhole endpoints lying about their neighborhood (scenarios (1) and (2) in section 4), the false positive rate do not exceed 8%. Yet, this proves the robustness and efficiency of the LNI's approach as for the detection of *simple* wormhole attacks. Our results discussed above show however that this approach is far from being sufficient, when wormhole attackers and their colluding nodes behave in a consistent way.

**SNV, Secure Neighbor Verification Protocol.** To evaluate the efficiency of the SNV protocol to our identified attacks, we plot in Figure 5 the true positive rates observed for several intensities of attacks. Again, as expected, the hit rate decreases as the population of malicious nodes increases. Indeed, we observe that the curves shape are slightly similar, whether the considered malicious nodes

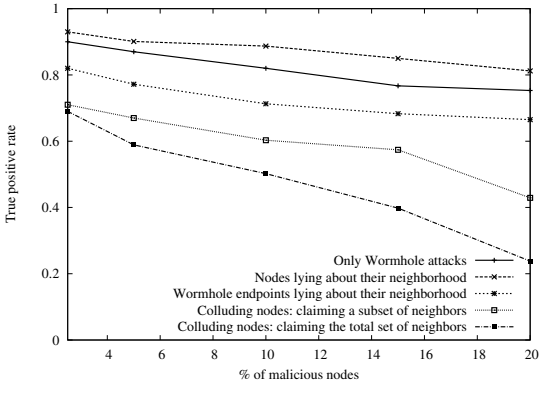


**Fig. 5.** SNV: True Positive Rate *vs* attacks intensity

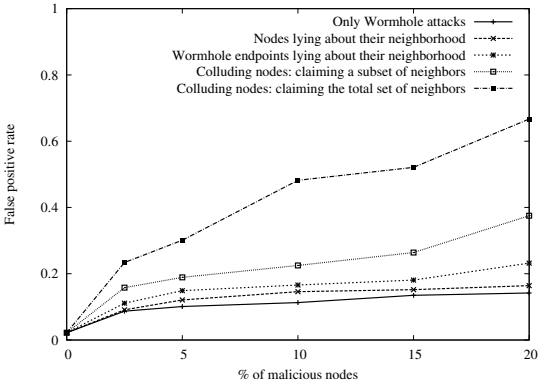


**Fig. 6.** SNV: False Positive Rate *vs* attacks intensity

are simple wormhole attackers, only lying about their neighborhood or colluding nodes. Surprisingly, we observed that when wormhole attackers simply lie about their neighbors (even without colluding with a set of neighbors), the scheme detects on average only 70% of malicious nodes, with a moderate intensity of the attacks (10% of malicious nodes). When faced to colluding nodes scenarios, and in particular when only a subset of neighbors lies in concordance with the wormhole endpoints, the detection ratio falls to a drastically low value (42% for 20% of malicious nodes inside the system). Recall from section 3 that an attack is detected only if the 4-clique is not a quadrilateral. However, biased lists of neighbors in SNV give malicious nodes opportunities to corrupt and distort network distances so that malicious nodes can form convex quadrilaterals which all their links are verified by the protocol. This explains the low detection rates observed for SNV in our different scenarios.

Figure 6 illustrates the variation of false positive rate as a function of the average percentage of malicious nodes in the network. We observe that the rate increases faster as the percentage of attackers increases. False positive rate is more important in the case of colluding nodes. Again, this is explained by the creation of additional links when liars send false lists of neighbors. Virtually created links in neighbors' exchanged tables cause the creation of false quadrilaterals that alter results of the quadrilateral test.

The salient result to be noticed finally is that in presence of colluding nodes, the detection scheme achieves worse results than a random detection. Indeed, merging both figures 5 and 6, we can evaluate the efficiency of the test as a ROC (Receiver Operation Characteristics) curve observed for several intensities of attacks. This shows that for a magnitude as low as 10% of attackers in the system, the SNV protocol behaves as a random detector when dealing with colluding nodes lying consistently.

In summary, even if the chosen protocol is specifically designed to secure neighbors discovery, it could not correctly detect partial or complete collusion of nodes whereas the first protocol seems to be more resistant to collusion.

## 6 Possible Countermeasures and Discussion

As shown in the previous section, the second method (SNV) which proposes a dedicated method for secure neighbors discovery is relatively inefficient when looking at colluding nodes whereas the first method (LNI) which is only designed for wormhole detection seems to behave better. More generally, geometric tests seem to be less efficient than local neighbors information combined with a voting method.

Notably, the attacks identified in the previous sections are generally applicable to other approaches detecting wormhole attacks or securing neighborhood information. Even if we only tested the impact of the attacks on two representative solutions, it is important to note that our attacks exploit a common vulnerability of all the methods proposed so far. Consistently lying about its neighbors remains the main weakness of previous solutions our attacks exploit to succeed. A malicious pair of nodes establishing a wormhole attack, *and* exploiting the knowledge of each wormhole endpoint's neighbors can easily mislead any technique that do not consider this as a possible threat.

A possible countermeasure could be to reinforce the first studied algorithm with a mechanism that do check whether the local neighbors information, even though consistent with other observations made by the testing node, is not deviant from observations as reported by a majority of other nodes. In other words, the LNI algorithm needs to be modified in such a way that the local neighbors information is verified with the help of a voting technique before the wormhole detection mechanism is activated.

Such mechanisms of neighbors list verification have been proposed in the literature for other purposes. In [18], the authors check the consistency of neighborhood tables between neighbor nodes to detect Sybil attacks. A Sybil node (i.e. a node with several identities) will declare several times the same set of neighbors leading to identities that appear many more times than others in the intersection of the neighborhoods. The same kind of mechanism could be used to detect inconsistency and thus liar nodes in a neighborhood tables verification mechanism: for a sufficiently dense network, the intersection of neighborhoods of a node's neighbors must be very close and must be consistent. This allows to detect liars with a higher confidence.

Moreover, as the motto of the wireless sensor networks could be "unity is strength", a voting mechanism could also be locally added as done in [6]. Indeed, those voting techniques stay efficient as soon as the number of attackers is less than 30% of the total number of nodes which is the most classical case.

An alternate solution would be to jointly run both mechanisms in an attempt to secure the neighborhood discovery process and detect potential wormhole attackers. However, two major reasons refrain the adoption of such an intuitive solution; First, the two classes of techniques have conflictual assumptions. The first technique relying on secure neighboring to detect wormhole, whether the second detects wormhole attacks assuming a secure neighboring state. Clearly, if one would run both solutions instantaneously, the assumptions of each are not verified. Second, running both solutions would obviously require higher energy

cost and would induce more overhead traffic. Notably, the latter reason is a major concern when adopting a solution consisting in simultaneously deploying any two techniques that would belong to each of the two classes.

In summary, a convenient countermeasure is to verify first the consistency of the different neighborhoods before running the wormhole detection test. The exchanged information and the energy cost of this consistency check is roughly the same than for the wormhole detection mechanism.

We have tested this approach that gather a test of neighborhood consistency and the LNI wormhole detection mechanism under the same simulation conditions than the previous ones. This leads to largely improve the previous results: when considering 10 % of colluding nodes that claim the total set of neighbors, the true positive detection rate of wormholes and of liars is always greater than 70 % whereas the false positive rate stays under 10 %. Those initial results seem to confirm that adding an initial consistency check of neighborhoods with a voting mechanism helps the algorithm to correctly detect wormhole endpoints even if they are hidden behind liar nodes. Of course, this initial study needs to be refined but these results are very promising.

## 7 Conclusion

In this paper, we analyzed the impact of simple attacks, providing biased lists of neighbors, on the protocols aiming to detect wormhole attacks and securing neighboring in WSNs. One of our salient findings is that, when wormhole attackers compromise a subset of neighbors so that these consistently lie about their neighborhood with the virtual topology created by the wormhole, the performance of the detection protocols can easily degrade below that of a random detector.

We have also described a possible convenient countermeasure and we presented the first encouraging results in this direction keeping in mind the motto of wireless sensor networks "unity is strength".

**Acknowledgment.** This work has been supported by the French ANR national project ARESA2. It has also been partially supported by the European Commission within the STREP WSAN4CIP project. We would like to thank the anonymous reviewers for their useful comments.

## References

1. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
2. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols 1(2-3), 293–315 (2003)
3. Hu, Y.-C., Perrig, A., Johnson, D.B.: Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In: Proceedings of the Twenty-Second Annual Conference INFOCOM, IEEE 2003, San Francisco, CA, vol. 3, pp. 1976–1986. IEEE, Los Alamitos (2004)

4. Hu, L., Evans, D.: Using directional antennas to prevent wormhole attacks. In: Proceedings of NDSS 2004, San Diego, California, USA. The Internet Society, San Diego (2004)
5. Čapkun, S., Buttyán, L., Hubaux, J.-P.: Sector: secure tracking of node encounters in multi-hop wireless networks. In: Proceedings of SASN 2003, pp. 21–32. ACM, New York (2003)
6. Znaidi, W., Minier, M., Babau, J.-P.: Detecting wormhole attacks in wireless networks using local neighborhood information. In: IEEE PIMRC, Cannes, French Riviera, France (September 2008)
7. Shokri, R., Poturalski, M., Ravot, G., Papadimitratos, P., Hubaux, J.-P.: A practical secure neighbor verification protocol for wireless sensor networks. In: WiSec 2009, pp. 193–200. ACM, New York (2009)
8. Maheshwari, R., Gao, J., Das, S.R.: Detecting wormhole attacks in wireless networks using connectivity information. In: INFOCOM 2007, Anchorage, Alaska, USA, May 6-12, pp. 107–115. IEEE, Los Alamitos (2007)
9. Khalil, I., Hayajneh, M., Awad, M.: Svnm: Secure verification of neighborhood membership in static multi-hop wireless networks. In: Proceedings of the 14th ISCC 2009, Sousse, Tunisia, July 5-8, pp. 368–373. IEEE, Los Alamitos (2009)
10. Poturalski, M., Papadimitratos, P., Hubaux, J.-P.: Towards provable secure neighbor discovery in wireless networks. In: Proceedings of the 6th FMSE 2008, Alexandria, VA, USA, October 27. ACM, New York (2008)
11. Brands, S., Chaum, D.: Distance bounding protocols. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
12. Liu, D., Ning, P., Du, W.: Attack-resistant location estimation in sensor networks. In: Proceedings of the Fourth IPSN 2005, UCLA, Los Angeles, California, USA, April 25-27, pp. 99–106. IEEE, Los Alamitos (2005)
13. Du, W., Fang, L., Ning, P.: Lad: Localization anomaly detection for wireless sensor networks. J. Parallel Distrib. Comput. 66(7), 874–886 (2006)
14. Wang, W., Bhargava, B.K.: Visualization of wormholes in sensor networks. In: Jakobsson, M., Perrig, A. (eds.) Proceedings of the 2004 ACM Workshop on Wireless Security, Philadelphia, PA, USA, October 1, pp. 51–60. ACM, New York (2004)
15. Poovendran, R., Lazos, L.: A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. Wireless Networks 13(1), 27–59 (2007)
16. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Science of the United States of America, PNAS 101(9), 2658–2663 (2004)
17. Hamida, E.B., Chelius, G., Gorce, J.-M.: Scalability versus accuracy in physical layer modeling for wireless network simulations. In: 22nd ACM/IEEE/SCS Workshop PADS 2008, Rome, Italy (June 2008)
18. Ssu, K.-F., Wang, W.-T., Chang, W.-C.: Detecting sybil attacks in wireless sensor networks using neighboring information. Computer Networks 53(18), 3042–3056 (2009)