Robust and Scalable Secure Neighbor Discovery for Wireless Ad Hoc Networks

Somayeh Taheri, Dieter Hogrefe

Institute of Computer Science, Georg-August University of Göttingen, Germany

Abstract—Several important network functionalities such as routing and localization rely on correct neighbor discovery. In wormhole attacks the adversary tunnels the packets from one to another area to mislead the nodes in those typically far away areas to believe to be neighbors. In this paper we propose Robust Secure Neighbor Discovery (RSND) which is a distributed neighbor discovery protocol to prevent such attacks. RSND leverages a combination of Multi-Dimensional Scaling and a loss function used in robust estimations to distinguish a node’s real neighbors from the ones induced by a wormhole. RSND constructs an estimate of the local topology using the measured distances between neighboring nodes such that the effect of wrong ranging values is down-weighted. This allows a node to distinguish which neighbors of it are introduced by a wormhole. We will verify the performance of our protocol using simulation study.

I. INTRODUCTION

In wireless networking neighbor discovery may affect critical procedures including routing and localization. In routing protocols the routing packets are forwarded by neighbors and in localization a node relies on its neighbors with known locations to localize itself. Node A is considered to be a neighbor to node B if A is within the radio range of B. Neighbor discovery is basically executed through exchanging Hello messages. Therefore, the open nature of wireless medium introduces a potential security threat to neighbor discovery mechanisms through giving the possibility of eavesdropping, replaying and disturbing such communication to attackers.

One of the sophisticated attacks against neighbor discovery is wormhole attack (relay attack). Typically a wormhole has two ends at two far away areas that (possibly selectively) tunnel packets between the two areas to mislead the nodes in one area to believe to be neighbors to those in the other. Consequently, the network layer functionalities may be disrupted and the attacker can monitor the communications between such nodes.

Secure Neighbor Discovery (SND) means to avoid attacks launched against correct neighbor discovery. Some approaches only detect if any wormhole is present while some others can also label the neighbors induced by the wormhole as incorrect. Correct neighbors mean the real physically neighbors of a node while incorrect ones are the apparently neighbors of a node introduced through wormhole connections.

Figure 1 shows a typical wormhole attack. If a pair of victim nodes at the two sides of the attack, let’s say A and B, try to measure the distance between them what actually they will measure would be the sum of their distances from the wormhole-ends, i.e. the range value affected by the wormhole will be

\[ D_{AB} = d_1 + d_2 + d_{wh} \]

where \( d_{wh} \) is due to the possible intended or unintended delay added during the tunnelling. This is because the ranging (distance measurement) transmissions must traverse the path A-WE1-WE2-B and vice versa.

In a graph consisting of a set of nodes and their ranging information, the changes to the distance values caused by the wormhole can be seen as inconsistencies. RSND detects the incorrect neighbors of a node by determining which edges in such a graph are the source of the inconsistencies.

The main contributions and advantages of RSND are as follows. In a distributed manner it performs secure neighbor discovery and verifies the correctness of the whole neighbors of a node at once to reduce the computational overhead. It requires only the current ranging values to check the neighborhoods. To the best of our knowledge, RSND is the first distributed range-based SND algorithm which takes the whole neighbors of a node at once and going beyond only detecting the presence of a wormhole attack detects which links are affected by the wormhole. One of the most important properties of RSND is that when no attack is present in a node’s neighborhood a significant part of the calculation overhead is avoided since detecting the existence of wormhole is separated from determining the affected links and when no attack is existing the second phase will not be executed by the nodes.

II. RELATED WORK

Some SND approaches measure the time-of-flight of the signal to measure the distance between nodes in order to verify neighborhoods, such as [5]. Some others use location information of the nodes, such as [8]. The drawback of the location-based solutions is requiring localization information of at least a subset of nodes.

In [9] ultra-sound time-of-flight ranging measurements are used along with some geometry tests. Neighboring nodes exchange neighbor tables together to build up a second order neighbor table including a list of their neighbors and the distances between each two nodes among them. In order to verify a claimed neighbor Y, a given node X will find two other neighbors such that they all are neighbors and the distances between all of them is available (i.e. they form a 4-clique). Then if these four nodes form a convex quadrilateral all links can be verified. The algorithm can not determine which link is inconsistent in...
a 4-clique. Another disadvantage of this work is that a quite high number of 4-clique tests must be done by each node.

A recent approach based on time and some topology tests is [10]. This is the first scheme which takes advantage of mobility for securing neighbor discovery. This protocol is designed for the situation when the nodes’ movement happens on straight lines (or other describable paths). This solution verifies the correctness of each neighbor separately. On one hand this is an advantage because it does not require the participation of any other node. On the other hand the algorithm needs to be executed once for each neighbor which may lead to a high calculation overhead beside the fact that it requires to measure the distance between the two nodes a couple of times while they are moving to observe the potential distortions made by the wormhole in the ranging information. A graph is formed whose vertices are the positions of the two mobile nodes at consecutive points of time and whose edges are their measured distances and the distances travelled by each one at each step. Such a graph is rigid given that the nodes move on straight lines (because in this case the distance travelled by a node between any two points can be calculated by adding the single values). The protocol first makes basic tests including to check the ranges for not being longer than the maximum radio range of nodes which is a sign of wormhole attack. If these tests are successful then Multi-Dimensional Scaling (MDS) method would be applied to the values in three iterations, to check if it is possible to fit the edge values to a rigid graph. The outcome of each MDS is fit to the known nodes’ path of travel (straight line). The goodness of this fit as well as the goodness of MDS of the best two among the three outcomes are used to decide about the presence of a wormhole. If the goodness values of them are worse than some thresholds a wormhole between the two nodes is detected. The results show that the enough number of ranging operations is around 12 for this method.

There are also approaches based on connectivity graph of the whole network. Such approaches are suitable for dense networks and look for distortions in the connectivity graph to detect the presence of wormholes. An example is [7] in which the distribution of node degree as well as the shortest path length is compared to the corresponding values of the correct network. Another work is presented in [13] and [12]. This approach inputs the connectivity information of the nodes to MDS method to estimate the virtual positions of the sensor nodes. Then if the resulted visualized network is bent to bring two far away points together a wormhole attack is detected. The drawback of this approach is its centralized nature.

Device fingerprinting as well as channel fingerprinting are the basis to some other existing approaches.

III. SYSTEM ASSUMPTIONS

Our system model assumes to have a number of ad hoc nodes distributed randomly in a 2D area. RSNd can be applied to both static and mobile scenarios as it requires the fresh ranging information in a node’s neighborhood which in mobile scenarios can be obtained once a while. Each node is equipped with a radio-frequency (RF) interface as well as an ultra-sound (US) interface which will be used for ranging operations with the transmission range of \( R \) regarding both RF and US. A node also needs a clock with the precision of hundred of microseconds for the timing purposes required for ranging. We use the ranging mechanism described in [9]. As required by this mechanism, the nodes must be able to do symmetric key encryption and decryption computations as well as message authentication and hash function computations. Each pair of nodes should have one shared secret key with each other possibly pre-loaded in advance.

IV. ATTACKER MODEL AND PROBLEM FORMULATION

We assume that the adversary has a number of wormhole-ends deployed in the network and would carry out wormhole attacks through them. The wormhole-ends, also known as relay nodes, have the same RF and US interfaces as in legitimate nodes and typically are far away from each other. One wormhole-end has the ability to eavesdrop messages heard in its vicinity and tunnel them, possibly selectively, to the other wormhole-end through an out-of-band connection (wormhole connection) possibly with the speed of light. So the relay nodes will have the possibility of relaying US messages from one point to another much faster than the normal speed of sound. The other wormhole-end would replay the messages to convince its neighboring legitimate nodes to be in the radio range of the far away legitimate nodes. Note that the attacker does not need to understand (decrypt) or modify messages, \( M \) is the number of nodes whose messages is tunnelled by a wormhole-end which is a random number possibly less than the network degree, \( N \) (the average number of neighbors of nodes). This is because if the wormhole-ends always tunnel all nodes in their range, the apparent node density around them will become twice the normal density of the network (legitimate nodes plus tunnelled ones) which can betray the presence of the attack. In the current version of this work we consider only two-end relay attacks. The attacker is not able to compromise nodes and his computational power is limited in the sense that he can not disclose encrypted messages.

The problem addressed in this paper is as follows: How a node can detect which nodes among its neighbors are really in its radio range and which ones are tunnelled to its vicinity by the wormhole attacks.

V. RSNd’s CONTRIBUTIONS

The existing SND protocols for static networks have some drawbacks as follows. [13] proposes a centralized approach to visualize the whole network at a central node to detect the sign of the wormhole. In [11] the test must be executed by a node for each neighbor separately. [9] detects if any wormhole attack is presented somewhere in the neighborhood relationships between a set of four neighboring nodes but can not determine which links are the incorrect ones. For mobile scenarios an interesting SND approach is proposed in [10]. It verifies the neighborhood of each pair of nodes separately and requires the mobility of the two nodes on describable path shapes while performing the ranging measurements.

Our protocol is a distributed approach and is able to distinguish correct and incorrect neighbors by testing the whole
neighborhood of a node together due to exploiting the robustness of a special form of MDS. It is also applicable to mobile applications without requiring mobility during the protocol execution or special movement patterns.

VI. RSND PROTOCOL OVERVIEW

RSND’s design is the answer to this question: when in a local set of nodes a number of links are affected by a wormhole attack, how to determine which nodes are the source of the inconsistencies in the ranging values. Figure 2 shows a node affected by a two-end wormhole attack. Remember that the measured distance between A and its incorrect neighbor is measured as mentioned in Figure 1.

![Figure 2](image_url)  
Figure 2. Node A as a wormhole attack victim and its correct/incorrect neighbors

We are interested in a method to configure the network topology locally (in a node’s neighborhood) using the available ranging values such that the effect of the wrong (inconsistent) distance values is down-weighted. The motivation is to design a distributed algorithm to find out who is on the wormhole rather than just is there any wormhole. The building blocks of RSND are Multi-Dimensional Scaling and robust statistics.

A. Main Idea

The main idea is to configure (visualize) the local topology of a node’s neighborhood based on the available ranging information in a robust manner such that the incorrect ranging values (the ones affected by the wormhole) affect the configuration less than the correct ones. Finally it will be possible to check which link values are 'ignored' by the visualization algorithm which are the candidates for incorrect links. Although MDS is a relevant tool for doing the topology configuration, it must be modified in order to satisfy the robustness property of our algorithm. In configuration of a set of points by MDS (detailed explained in section VI-B) the goal is to find the relative positions of the points while preserving the given distances as far as possible. This goal, as described later, is realized by minimizing the sum of the residuals (differences between input and output distance values). We would like to modify this goal to minimizing the sum of ‘weighted’ residuals in which the residual of incorrect links are down-weighted. The algorithm thus would intend to preserve the ranging values of correct links while letting the wrong ranging values be much more free to change because of having a down-weighted role in the error function. After doing such a visualization, we will be able to detect the wormhole affected links by checking which links have changed more significantly.

We would call the MDS link errors, i.e. the difference between the measured distances and the corresponding values turned out from MDS, as "link wormhole sign value"s. We need to modify MDS to be robust to the inconsistent ranges and then to detect the links with the high "link wormhole sign value"s which would be candidates for wormhole affected edges.

For the purpose of robustness in MDS we use the approach proposed by Heiser in [4] in 1987. In that paper Heiser proposes to use Huber criterion in the loss function of MDS to weaken the effect of more erroneous edge values (in our work it would be the links touched by the wormhole attackers). One important property of Huber function is being independent from the underlying error distribution. Heiser proposed a robust loss function for MDS which is also straightforward to be minimized as will be described in section VI-C.

B. Building Blocks

Robust Statistical Estimation: Classical statistics develop estimation methods assuming that the data errors have a normal distribution. Ordinary (linear) Least square estimation (OLS) method estimates the data values such that the sum of squared residuals, i.e. squared differences between the predicted values and their corresponding observed values in the dataset, (known as cost function) is minimized. But in practice not the whole data fits to the same model, as in presence of outliers. Robust statistical estimators would much better fit such situations. Robust M-estimators are a good example of robust functions to make the procedure insensitive to the deviations from the underlying assumptions. The idea is to modify the estimator function such that the terms related to large residuals are down-weighted compared to the small ones.

Two very common loss functions are $L_1$ and $L_2$ norms. In $L_2$ norm, the residual $r_i$ takes part as $r_i^2$ in the cost function which would form the Ordinary Least Square estimator. $L_1$ is a simple robust M-estimator, which is also known as Least Absolute Estimator in which the residual $r_i$ would appear as $|r_i|$ in the cost function. $L_1$ norm decreases the sensitivity of the estimator to the outlier values simply because is increases slower for large residuals in comparison to $L_2$. Huber loss function [6], as a practical alternative robust loss function, can be seen as a combination of both mentioned norms. It is quadratic for small residuals and uses the absolute value for the large ones, as described in section VI-C. One of its important properties is being continuous.

Multi-Dimensional Scaling: MDS is a method to find a spatial configuration of the objects (in 2D or higher dimensions) based on the given information about their pairwise distances (usually referred to as dissimilarities). Classical and Metric MDS get the dissimilarity matrix (e.g. Euclidean distances between objects) and output a coordinate matrix such that the loss function is minimized. Most common form of the loss function is Least Square of residuals. Metric MDS gives the possibility of considering every dissimilarity value with its own weight in the MDS mechanism.

In general the dissimilarities are presented as a symmetric matrix, $D$, with zeros along the diagonal. The coordinate matrix, $X$, as the output presents the calculated position of each object in one row. Let $d_{ij}$ be the given dissimilarity between objects $i$ and $j$ in $D$ and $d_{ij}'$ be the estimated distance between
objects $i$ and $j$ based on $X$. A very well-known method to solve MDS problem, i.e. to find $X$ such that the loss (or stress) function as shown in equation 1 is minimized, is majorization.

$$\sigma_r(x) = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij}(X))^2$$  \hspace{1cm} (1)

The principles of majorization is as follows. When the function, $f(x)$, is not straightforward to be minimized through derivation if we can find another function, $g(x, z)$, such that 1) it is simple to be minimized and 2) for every constant $z_0$ we have $g(x, z_0) \geq f(x)$ and also $g(z_0, z_0) = f(z_0)$, then $f(x)$ can be minimized through iteratively minimizing $g(x, z)$.

For the latest purpose we would first choose a start point $x = z_0$ and find the minimum of $g(x, z_0)$ over $x$ which let’s say happens at $x = z_1$. Based on the properties of $g(x, z)$ we know that $f(z_1) \leq g(z_1, z_0) \leq g(z_0, z_0) = f(z_0)$. Now if we continue with finding the minimum of $g(x, z_1)$ over $x$ which would happen at let’s say $x = z_2$ in the same way we will have $f(z_2) \leq g(z_2, z_1) \leq g(z_1, z_1) = f(z_1)$. This means the value of $f(x)$ at the minimum of $g(x, z_i)$ is getting smaller in each iteration (unless $f(x)$ is not bounded from below at all). Therefore, in each iteration we are getting closer to the minimum of $f(x)$ and if the algorithm continues until $f(z_{i-1}) - f(z_i) < \epsilon$ for a very small positive $\epsilon$ it will end up minimizing $f(x)$. Majorizing the stress function as given in equation 1 goes back to [2] and [3]. We refer the reader to [1] for the mathematical details of this majorization mechanism.

C. The Scheme

As mentioned before, Heiser proposed to use Huber function to make MDS robust against observation errors in dissimilarity values. He developed a majorization algorithm for such a minimization problem and called the robust stress function as Huberized loss function [4]. We exploit Huberized MDS in order to detect wormhole attacks in neighbor discovery. Our protocol consists of the following steps.

1) Ranging and Exchange of ranging information: The ranging operation can be performed at the initiation phase of the network and then triggered when the network topology changes. Our protocol can be run at any time that a fresh setting of ranging information is available even only in the local neighborhood of one node. Any ranging method can be used for instance, the one using Ultra-sound time-of-flight similar to [9] as described below briefly.

When node $A$ decides to measure its distances to its neighbors it sends a REQ messages to each apparently neighboring node. The REQ message sent to the neighbor $B$ contains a nonce, $N^a_B$, encrypted with the key shared between $A$ and $B$ which will be used later to authenticate $B$ and also the hash value of another nonce, $N^b$ which is the same for every neighbors. The whole message is authenticated with a MAC to $B$ who will reply with a REP message containing $N^b_A$. $A$ and $B$ record the time of receiving and sending the REQ and REP as $t^A_{REQ}$, $t^B_{REQ}$, $t^B_{REP}$ and $t^A_{REP}$, for synchronisation purpose.

Then $A$ broadcasts an ultra-sound ranging message to all of its neighbors. Again both $A$ and $B$ record times of sending or receiving this message $t^A_{RNG}$ and $t^B_{RNG}$.

In the last step, $A$ sends acknowledgements to the neighbors who had sent back a correct REP. This message contains the timings recorded at $A$ for the corresponding neighbor as well as $N^b$. Node $B$ compares the $N^b$ received in this step with the one stored before and if they are the same it will be possible to check if there is any delay beyond the propagation delay in its transmissions to $A$. It checks weather the sum of RF latencies (from $A$ to $B$ in REQ plus from $B$ to $A$ in REP) is less than a threshold. Note that if there is any difference between $A$ and $B$’s clocks it will be cancelled in this summation. If there has been no delay beyond the RF propagation delay (which is in order of microseconds), node $B$ will calculate its distance to $A$ as $d_{AB} = \frac{1}{2}(t^B_{RNG} - t^B_{REP} - (t^A_{RNG} - t^A_{REP})) \times s$ where $s$ is the speed of sound. In calculating $d_{AB}$ the RF propagation delay is subtracted from US propagation delay in order to compensate the clock difference between $A$ and $B$ if there is any.

After ranging, each node broadcasts its measurements to its neighbors. If a node has discovered $N$ neighbors in the ranging phase it would create a $N \times N$ symmetric dissimilarity matrix, $D$, whose $(i,j)$th entry is the distance between its $i$th and $j$th neighbors. The node inserts itself on the first column and row of the matrix, i.e. fills them with its own measured ranges to its neighbors. Note that to fill any other column and row the node needs to use the ranging information received from neighbors. Obviously, some entries of the matrix are unknown because some neighbors of the node might be out of each other’s radio range.

2) Wormhole presence check: After creating $D$, a node would perform security checks possibly in two steps as follows. Step 3 will be executed only if the result of Step 2 (either in part a) or part b) as follows) is positive.

a) First, the node as a verifier makes checks if any entry of $D$ is larger than $R$. If so, it will set its wormhole flag up and will go directly to step 3 and otherwise to part b of this step.

b) The node inputs matrix $D$ as the dissimilarity matrix to metric MDS algorithm and calculates the stress value. If the stress value is below a threshold, $\tau_w$, (no inconsistencies) non of the node’s neighbors are induced by a wormhole attack and the protocol terminates at this point. Otherwise, the node will set its wormhole flag up and will go to step 3.

3) Detecting incorrect neighbors: Since the node’s wormhole flag is already set up, indicating that a wormhole attack is present, it would be responsible to verify . which neighbors are incorrect. This will be done by running the robust MDS method on matrix $D$ of the verifier node as described below.

Robust MDS algorithm: Huber function is defined as [6]:

$$H(r) = \begin{cases} r^2 & |r| < k \\ 2k|\vert r \vert - k^2 & |r| \geq k \end{cases}$$  \hspace{1cm} (2)

By replacing Least Square loss function with Huber function in Equation 1 the loss function will look like:

$$\sigma(X) = \sum_{i<j,r_{ij}(x)<k} w_{ij}r_{ij}^2(X) + \sum_{i<j,r_{ij}(x)\geq k} w_{ij}(2k|r_{ij}(X)|-k^2)$$  \hspace{1cm} (3)

in which $r_{ij}(X) = (\delta_{ij} - d_{ij}(X))$. This way, the role of large residuals in the loss function would be weakened. Therefore,
the small residuals are the dominant consideration in finding the configuration which minimizes the stress.

The majorizing function for this new form of stress function is proposed by Heiser in [4] as follows.

$$
\sigma_{maj}(X, Z) = \sum_{i<j, r_{ij}(Z) < k} w_{ij} r_{ij}^2(X) + \sum_{i<j, r_{ij}(Z) \geq k} k w_{ij} r_{ij}^2(Z) - k^2 w_{ij} 
$$

(4)

For the proof that $\sigma_{maj}(X, Z)$ majorizes $\sigma(X)$ we refer the reader to [4]. Now the procedure explained in VI-B must be followed to minimize $\sigma(X)$ using this majorization function. In each iteration with a fixed $Z = Z_i$ the task of finding the minimum of $\sigma_{maj}(X, Z_i)$ leads to an ordinary MDS minimization problem which is a known problem. This is because Equation 4 for a fixed $Z$ has the same form of the original loss function of MDS as presented in equation 1.

Our algorithm applies robust MDS method to matrix $D$ to find the configuration of the neighbors of the verifier node based on Huber loss function. Consequently, the erroneous ranging values will have a down-weighted role in configuration.

Then, as mentioned before, regarding the robustness of the algorithm the bigger a wormhole sign value (i.e. the difference between a link’s input and output ranging values) the higher the likelihood that a wormhole has affected that link. We call the matrix of the whole sign values sign matrix. The unknown entries of $D$ would have $w_{ij} = 0$ in the above procedure.

**Analysing sign matrix to detect incorrect neighbors**

Based on analysing the sign matrix we would classify a node’s neighbors into two groups: correct and incorrect neighbors. Note since the links between any two correct or any two incorrect nodes are not affected by the attack, they are expected to have small sign values in contrast to the links connecting a correct node to an incorrect one. Based on this, to distinguish the two groups we look for the two different patterns of sign values in the matrix. We do it by calculating the product of the whole entries in each column of the sign matrix (which is related to one particular node) and classifying the results into two classes based on their magnitude using a classification function in Matlab. It is worth mentioning that to calculate the sign matrix there is a problem due to the unknown entries in $D$ matrix. We solved this problem by estimating the missing values in $D$ before creating the sign matrix. For this purpose, we first apply the ordinary MDS to $D$ setting the MDS weights of missing values to zero and then we use the output configuration as an estimation for $D$ (including the missing ones) in the case that the stress value of this MDS function is lower than a threshold (which indicates the high correctness of the estimation) to create the sign matrix (otherwise only the available dissimilarity values would be considered).

**D. Discussion: Complexity**

We briefly discuss the complexity of RSND and compare it to [10]. In RSND each node first needs to perform MDS on its neighborhood ranging information. The computational complexity and memory requirement of metric MDS is $O(N^2)$ where $N$ is the number of points. Therefore, in this phase the computational complexity of RSND would be in the order of $SN^2$ where $N$ is the network degree (because number of points input to MDS is the neighborhood size) and $S$ is the network size. Then the nodes who detect the existence of a wormhole attack would perform a robust MDS. Let’s assume that $w$ is the fraction of the nodes who detect an attack. Thus, the computational overhead in the second phase of the protocol would be in the order of $wSIN^2$ where $N' = N + M$ is the average neighborhood size of wormhole affected nodes. Based on the reason discussed before, $M$ is a random number less than $N$, let’s assume $1 \leq M \leq N$. So the average value of $N'$ would be $\frac{2}{3}N$. Consequently, the whole computational complexity of RSND would be $C_{RSND} = \frac{2}{15}wSIN^2 + SN^2 = SN^2(1 + \frac{2w}{15})$. Our simulation results suggest that the average number of iterations in a robust MDS is around and usually less than 20 (in the scenario of Figure 5 on average we had $I = [14, 20, 18, 20, 18, 19]$ for $N = [8, 10, 12, 14, 16, 18]$). If we also assume $w = 20\%$ then $C_{RSND} = SN^2(1 + \frac{20}{15} \times 0.2 \times 20) = 7.25SN^2$.

In [10] each node must repeat MDS 3 times for each neighbor over 12 ranging which translates to 24 points. As the advantage of verifying the neighborhood in a pairwise manner, it can be assumed that for a pair of neighboring nodes it is enough if only one of them performs the protocol and informs the other one about the result. Therefore, the whole complexity of that protocol will be in the order of $\frac{3SN(24)}{2} = 3 \times 24 \times 12 \times SN$.

With $N = 12$ we will have $C_{RSND} = 87 \times 12S$ and therefore the computational complexity of RSND will be 10 times less than [10]. Another advantage of RSND is that it only requires to perform the ranging operations once.

Furthermore, it may not be necessary to fully minimize the majorizing function (regarding the number of iterations) in RSND. So one could try to tune the algorithm by experimenting with the number of (sub)iterations, which we plan to investigate in an extended version of this work.

**VII. Performance Evaluations**

In our simulations nodes are uniformly distributed in a 2D field having radio (and ultra-sound) range of $R = 300$m. The ranging error has a Gaussian distribution with the variance of $e$. In each set-up there is a single two-end wormhole attack located in arbitrary places in the network. First, we evaluate the correctness of the algorithm in determining if there is any wormhole attack in a node’s neighborhood. The evaluation criteria are false-positive and false-negative metrics. We investigate the effect of ranging error standard deviation from $e = 0$ to $e = 2.5$. The wormhole detection threshold, $\tau_w$, is increased accordingly as mentioned in the Figure 3. According to Figure 3, with larger ranging error the false-negative rate shows a very slight change while the false-positive rate grows more significantly. This is because a high ranging error itself can introduce inconsistencies to the ranging information of a correct set.

The second part of simulations investigates the performance of RSND in detecting the correct/incorrect neighbors of a wormhole affected node. First with $N = 12$ we increase the
number of nodes tunnelled by the wormhole as $M = 1$ to $11$. We run the experiment in the neighborhood of 80 nodes for each $M$. The result is shown in Figure 4 in terms of the accuracy of detections. As can be seen, by increasing $M$ for a fixed $N$ the accuracy of the algorithm degrades. This is because with a fixed $N$ as $M$ grows the wrong part of the ranging information grows while the consistent part is kept fixed.

Figure 3. Accuracy of RSND in detecting the presence of wormhole for $N=11$, $\tau_w=[10^{-14},0.005,0.03,0.06,0.12,0.15]$ is used for $e=[0,0.5,1,1.5,2,2.5]$

Figure 4. Accuracy of RSND in determining correct and incorrect neighbors of a node for $N=12$ ($e=1$)

Now based on the reason described in section IV we let $M$ be a random number in $1 \leq M \leq \frac{N}{2}$ to simulate a more realistic scenario. With random $M$ values and running over 150 neighborhoods, we got the results presented in Figure 5. It shows that the performance is not significantly affected by the network degree and it even improves slightly with $N$. This is because although $M$ is chosen to be proportional to $N$ and thus the ratio of true and wrong ranging informations does not change, as $R$ is fixed, for larger $N$ values more nodes are located in one node’s radio range (higher node density) and therefore more neighbors of the verifier node would be neighbors themselves and can measure their distances. Thus, the ratio of unknown edges would be less and with such a richer ranging information the topology could be configured more correctly by the algorithm. This indicates the very high scalability of RSND protocol. Figure 5 also shows the effect of $e$ which is obviously as expected.

VIII. CONCLUSION AND FUTURE WORK

A new secure neighbor discovery protocol for ad hoc networks based on MDS and robust statistics was proposed. RSND uses local ranging information in a distributed manner and analyses that to detect if there is a wormhole and further to ‘identify’ the incorrect neighbors. Using simulations, we showed the good performance of RSND for different network degrees (high scalability) and ranging error values. Although RSND uses MDS in iterations, its complexity is even better than similar SND methods which makes it suitable for ad hoc and sensor networks. This is because RSND involves the computation of detecting incorrect neighbors only if the presence of a wormhole is detected and additionally the verifications are performed at once on the whole neighborhood of a node. The future work would concern improving the performance of RSND and considering multi-end wormholes as well as using this work in combination with a localization protocol. Also, we plan to investigate if it would be possible to reduce the complexity of RSND by minimizing the majorization function only with a number of sub-iterations.

REFERENCES