SECUND: A Protocol for SECUre NeighborhooD Creation in Wireless Ad hoc Networks

Thaier Hayajneh
University of Pittsburgh
Pittsburgh, PA, USA
Email: hayajneh@sis.pitt.edu

Prashant Krishnamurthy
University of Pittsburgh
Pittsburgh, PA, USA
Email: prashant@sis.pitt.edu

David Tipper
University of Pittsburgh
Pittsburgh, PA, USA
Email: dtipper@sis.pitt.edu

***Abstract*—The ability to correctly determine their neighborhood is a fundamental requirement for nodes in ad hoc and sensor networks. Many applications, protocols, and system functionality rely on neighborhood discovery. Malicious nodes that taint neighborhood information using wormholes can significantly disrupt the operation of ad hoc networks. Protocols that depend only on cryptographic techniques (e.g, authentication and encryption) may not be able to detect or prevent such attacks. In this paper we propose SECUND, a protocol for creating a SECUre NeighborhooD, that makes use of discrepancies in routing hop count information to *detect* "true" neighbors and *remove* those links to nodes that appear to be neighbors, but are really not neighbors. SECUND is simple, localized and needs no special hardware, localization, or synchronization. We present approaches to improve the efficiency of the process. We evaluate SECUND using simulations and we demonstrate its effectiveness in the presence of multiple and multi-ended wormholes.**

I. Introduction

Nodes in ad hoc networks typically try to discover their neighbors simply by broadcasting a neighbor discovery request. Each node that hears the request responds with a neighbor discovery reply. An adversary may try to thwart neighborhood discovery to disrupt the network operation by (a) preventing neighbors from discovering each other by jamming or (b) creating a "neighbor relationship" between nodes that are not really in range of each other. The latter can be accomplished by spoofing neighbor discovery messages or by installing wormholes [1] in the network. Cryptographic techniques (authentication and encryption) can often prevent the adversary from spoofing the discovery messages. Wormhole attacks, considered in this paper, cannot be addressed using cryptography. Jamming attacks are not considered in this work.

A wormhole (see Figure 1) can be constructed by an adversary by simply copying all packets (signals) from one location (M_1) in the network and replaying them at another location (M_2) that is located several hops away. All the reply packets (signals) from location (M_2) will also be captured and replayed at location (M_1). Since the adversary can capture the signals or bits, cryptographic techniques by themselves will not help. Consequently, nodes that are located in M_1's range (e.g., A and D) will believe that they are neighbors to nodes that are located in M_2's range (e.g., B and F). In effect, the wormhole has created several bogus "direct" links between nodes in the network. Multiple wormholes and multi-ended wormholes can worsen the situation.

A short survey of neighborhood discovery in ad hoc networks is provided in [2]. A conclusion in this paper is that securing the neighborhood discovery process is a difficult and open problem. Further, the proposed protocols/algorithms (described in Section II) must be applied between *all pairs of nodes* to detect the existence of a wormhole. In large networks with high node degrees, this can result in significant overhead and delay. Finally, detection of the existence of a wormhole is not sufficient. It is necessary to correctly identify the bogus links and distinguish them from real links between neighbors.

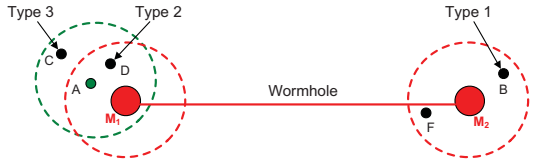

Fig. 1. Wormhole and Types of Neighbors

In this paper we propose SECUND [1], a protocol for creating a SECUre NeighborhooD that can discover true neighbors, distinguish between true and purported neighbors, and detect and remove wormhole links if they exist in the network. Compared to other secure neighbor verification or discovery protocols, SECUND is simple, localized, needs no special hardware, localization, or synchronization. SECUND is based on principles developed in [3] where we presented a protocol called DeWorm to detect the existence of wormholes. DeWorm is an on-demand protocol that makes use of routing hop count discrepancies, determined by nodes along a route in a sliding fashion, to detect wormholes that may be somewhere along a route. SECUND also makes use of routing hop count discrepancies, but its goal is to efficiently check links between every pair of nodes for existence of wormholes and to remove only tainted links to the extent possible. SECUND can also detect and remove two-ended and multi-ended wormholes (not considered in [3]). It has an excellent detection rate as shown by simulations in a variety of scenarios.

Using hop count discrepancies to detect the existence of a wormhole, independently by all nodes, results in an increased number of false positives. We show by simulations and arguments that the number of false positives can be significantly reduced by *mutual* checks for existence of wormholes between

[1] In botany, the word "secund" refers to having elements on one side only – for example, leaves on one side of a branch – and not on both sides. The SECUND protocol ensures that the final list of neighbors of a node are those on the same side as the node, not the other side of a wormhole.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

pairs of nodes. We also show that the number of such checks can be drastically reduced (without significantly impacting the detection rate or false positives) when nodes follow specific rules that enable them to *omit* such checks. When a wormhole is detected in the vicinity of a node, removing only those bogus links that have been created by the wormhole is challenging. SECUND is able to distinguish between different types of neighbors (see Figure 1) to remove bogus links. Mutual checks can reduce the number of legal links removed.

To the best of our knowledge, this the first protocol that employs cooperation between honest nodes to reduce the overhead associated with the number of checks to be performed to detect the existence of wormholes and create a secure neighborhood. This is also the first protocol, to our knowledge, that *explicitly* addresses the removal of bogus links without removing legal links. Our simulations show that the proposed protocol can successfully detect and remove most wormhole links. Very few legal links are mistakenly removed. The cost associated with SECUND is the overhead measured by the number of route acquisitions (which we show is fairly low and only involves discovering very short routes).

The rest of the paper is organized as follows. In Section III, we describe SECUND in detail including the mutual cooperation between nodes to reduce the overhead of securing the neighborhood. Section IV presents the results of the performance evaluation. Section V concludes the paper.

II. Related Work

A survey of neighborhood discovery and neighborhood discovery protocols are presented in [2]. According to [2], researchers consider any wormhole defense mechanism as relevant to secure neighborhood discovery. Thus, here, we will classify protocols for wormhole detection based on the approach they rely upon (even if they do not explicitly consider secure neighborhood discovery).

In location-based approaches, the location-aware sender and the receiver will securely exchange their location information. Then, to detect whether a wormhole connects them, the nodes will check if packets have traveled the distance between them using only a few hops and/or in a short time. In [4] end-to-end wormhole detection is proposed. Based on geographic information exchanged, the source node estimates the minimum hop count to the destination. The source compares the hop count value received from the reply packet with this estimated value. If the received value is less than that estimated, the corresponding route is marked as if a wormhole exists. Hu et al. [1], [5], suggested the use of geographical leashes to detect wormholes. A geographical leash requires each node to know its own location and all nodes to have loosely time synchronized clocks. The nodes need to securely exchange location information. A sender node can then ensure that the receiver is within a certain distance and detect discrepancies therein. Location based protocols usually require the nodes to be equipped with GPS, which may still not provide the required location accuracy (e.g., indoor and urban areas).

Time-based protocols, in general, are based on accurate time measurements or require the nodes to have tightly synchronized clocks. The idea here is that an in-band wormhole must cause noticeable delay for the traffic that passes through it. For instance, in [6], timing associated with existing MAC layer acknowledgments are used to detect a wormhole. In [7] authors proposed a transmission time based mechanism (TTM) to detect wormholes. The protocol requires the computation of the transmission time between every two successive nodes along the established path during route setup procedure. Time based protocols require some approximations as the node that is in charge of detection has to account for the processing and propagation delay times. Moreover, in ad hoc networks, the MAC protocol may also cause some unpredictable delays. More importantly, these protocols are not capable of detecting out-of-band physical layer wormholes. In [8], researchers showed that it is impossible to secure the neighborhood with general time-based protocols if adversarial nodes are able to relay messages with a delay below a certain threshold. Note that this threshold is what is typically used by such protocols to detect wormholes. A similar conclusion was also reached by Chiang et. al. [9].

Distance bounding approaches use estimates of the physical distance between purported neighbors to ensure that it is not longer than the maximum allowable distance (e.g., farthest distance reachable by a node operating at its maximum transmission power). Many techniques have been used to estimate the distance between the nodes. Some researchers relied on the signal round trip time and multiplying it by the signal propagation time (speed of light) [2]. A secure neighbor verification protocol for wireless sensor networks is proposed in [10]. Their protocol is distributed and relied on the estimated distance between nodes. They require each node to be equipped with a microsecond precision clock and two network interfaces: a radio-frequency and a sound interface. Other approaches also use some special hardware such as directional antennas [11], special RF [12], or ultrasound [13] to estimate such distance bounds. These protocols cannot be easily applicable to any ad hoc network because they add expense, complexity, and need for special customization. Moreover, some of these protocols have their own specific weakness and cannot always ensure the detection of wormholes. Also it is possible for the attacker to use adversarial nodes that are equipped with the same hardware used by the network nodes. For example, an attacker could also use directional antennas and align them in a way to deceive the detection protocol.

Protocols that do not rely on location, timing, or tight synchronization can be classified into centralized and distributed approaches. Centralized approaches rely on gathering information such as statistics and visual analysis on the network connectivity graph and processing them at a central entity. In [14], the network is reconstructed using multi-dimensional scaling and a wormhole that exists is detected by visualizing the anomalies introduced by the attack. Poovendran and Lazos [15] presented a graph theoretic framework for modeling wormhole links and derive the necessary and sufficient conditions to detect and defend against wormhole attacks. Qian et al. [16] presented a scheme to detect wormhole attacks based on statistical analysis. It is always preferred to have the process for detecting and removing wormholes decentralized or distributed in ad hoc networks – centralized approaches are

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

not very attractive. Centralized topology information was used in [17] to detect wormholes. The protocol looks for forbidden substructures in the connectivity graph that should not be present in a legal connectivity graph. This also requires the network to be highly connected. Detection requires a specific number of independent neighbors for the nodes connected through a wormhole to exist.

Decentralized or distributed approaches include protocols that are based on connectivity and neighborhood information. These are the closest in scope to SECUND – our proposed protocol. Here nodes will exchange information such as node degrees or the list of one-hop and/or two-hop neighbors. Based on the collected information, the existence or not of a wormhole is determined. The node degree is used to detect wormholes in [18]. The assumption here is that the wormhole will increase the number of one-hop neighbors of a node and if this number is greater than some threshold (e.g., the average node degree) then there must be a wormhole. If however the wormhole connects a single node with another node that is far away, the node degree only changes by one and the wormhole will not be detected. Another possibility could be to place the wormhole between nodes that have a node degree less than the average which will prevent the wormhole's detection. But the damage to the network is comparable to any other wormhole. The protocol suggests an approximate removal process for a set of suspicious links that may however completely isolate some nodes from the network. In [19], an approach similar to [18] was presented. Again the assumption made is that the wormhole will significantly increase the number of one-hop neighbors. Each node will count the number of nodes that are two-hops away and the idea is that this number grows under a wormhole attack. In [20], the network topology is assumed to be static, links are assumed to be bidirectional, and in a dense network every two neighbors are assumed to have a common neighbor. They assumed that the wormhole must change the topology structure of the network and they computed a so-called edge-clustering coefficient. A wormhole node is detected by one of its neighbors if that neighbor cannot reach one of the wormhole neighbors without using that node. However, it is very possible to come up with many scenarios with wormholes that will not satisfy any of the necessary conditions with this approach to detect the wormhole. This will only successfully detect open wormholes or closed wormholes that only connect one single node with another single node. If the wormhole connects a group of nodes (≥ 2) with another group of nodes, which is the most common form of wormhole, then the protocol will not detect the wormhole.

III. SECUND

In this section, we describe how SECUND works. We will first describe the *detection* of the existence of a wormhole without regard to false positives or the number of checks. Then we will consider the use of *mutual* checks to reduce false positives and the use of *rules* to allow certain nodes to omit checks for wormholes. Finally, we discuss the problem of *identifying types of neighbors* and *removal* of bogus links.

A. Network Model and Notation

Consider an arbitrary ad hoc or sensor network consisting of n nodes represented by the ordered set Q. Let the set of one-hop neighbors of a node A be N_A, that is, $N_A = \{A_1, A_2, ...A_{k_A}\}$, where k_A is the number of neighbor replies received by node A. For the discussion in this section we will assume the existence of a single two-end wormhole. The wormhole equipment $M_1 \leftrightarrow M_2$ is defined as two extra nodes M_1 and M_2 that are not part of the network, i.e., not elements of Q. Here we assume a closed wormhole where M_1 and M_2 are not visible to their neighbors (i.e., they do not advertise their node IDs or MAC addresses) and that the wormhole is an out-of-band physical layer wormhole that uses a high speed link to connect M_1 and M_2. Detecting such wormholes is considered to be extremely difficult [21]. The set of one-hop neighbors of M_1 and M_2 will be N_{M_1} and N_{M_2}, respectively. Note that by definition, every node in N_{M_1} is connected to all the nodes in N_{M_2} via the wormhole and vice versa. Thus N_A, the one-hop neighbor set of node A includes nodes both within transmission range and on the other side of the wormhole if A is in the transmission range of the wormhole. Let $\hat{N_A}$ be the set of "true" one hop neighbors of A. Then $N_A^* = N_A - \hat{N_A}$ will be the set of nodes that are not true neighbors of A. Clearly, $N_A^* = N_{M_2}$. With reference to Figure 1, $N_A = \{B, C, D, F\}$ and $\hat{N_A} = \{C, D\}$ and $N_A^* = \{F, B\}$ (Type 1 neighbors). The set $\hat{N_A}$ comprises of nodes that may also belong to N_{M_1} – these are called Type 2 neighbors (e.g., D) and nodes that are not in N_{M_1} like C that are called Type 3 neighbors. Let the route from any node X to any node Y be R_{X-Y} and $|R_{X-Y}|$ be the length of the route in number of hops.

B. Detection of Existence of Wormhole

Node A will first determine if it is in the vicinity (within the transmission range) of a wormhole. The process used here is similar to that described in [3]. The basic idea here is as follows. If node A is in the vicinity of a wormhole, one or more nodes in N_A will be on the other side of the wormhole. Suppose that $B \in N_A^*$ is not a true neighbor of A and that the wormhole is η hops long. If a node $X \in \hat{N_A}$ were to find a route to some neighbor of B that is not a neighbor of A (called the *target* T) avoiding all nodes $\in N_A^*$, such a route must be at least η hops long (since a route that goes through the wormhole has to include some node in $N_A^* = N_{M_2}$, the wormhole is avoided and the alternate route must be at least as long as the wormhole itself). Implementing this idea is not trivial since node A does not know the composition of N_A^*. So node X avoids using all nodes in N_A which will include all nodes in N_A^*. But X itself may be part of N_A^* making it necessary for all nodes in N_A to repeat this process. Further, if $T \in N_{M_1}$, it could be closer to A than B. All of these are taken into account in the algorithm to detect existence of a wormhole shown in Figure 2. A detailed description of all the steps with some discussion is presented next.

1: Node A will discover its one-hop neighbors by broadcasting a "hello" message. Cryptographic techniques (e.g., authentication) are used to prevent malicious nodes from sending fake replies.

2: Node A receives replies from its neighbors and verifies their

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

authenticity. Neighbors could be elements of $\hat{N}_A$ or N_A^*.
3: Node A wishes to determine if B is a true neighbor. A asks B to provide its one-hop neighbor list N_B. We refer to B as the neighbor under examination.
4: Node A picks some node $\in N_B - N_A$ and marks it as the target node T.
5: Node A will ask all its one-hop neighbors (real and purported) to find the shortest route to T. Those routes: (i) cannot be direct (must pass through another node) and (ii) must avoid the one-hop neighbors of both A and B.
6: Nodes in N_A reply to A with the length of their shortest routes to the target node T.
7: Node A employs *Select(route)* (see Figure 3) to select a route that will be compared with the wormhole route (the wormhole route should be 3 hops – from the neighbor of A to A, from A to B and from B to T). Using *Select(route)* eliminates extremely long route outliers while ensuring that a route that is η hops longer than the wormhole route is not missed.
8: If the difference between the length of the selected route and the wormhole route is greater than the wormhole length η then the neighbor under examination is connected by a wormhole link.

Detection(A,B)
1. A **finds** N_A
2. B **sends** N_B to A
3. A **chooses** $T \in \{N_B - N_A\}$
4. $\forall X \in N_A$, X **finds** $R_{X-T} : N_B \cup N_A \not\subset R_{X-T}$ and $|R_{X-T}| > 1$
5. $\forall X \in N_A$, X **sends** $|R_{X-T}|$ to A
6. A **determines** *Select*(R_{X-T}) and **computes** $|R_{S-T}|$
7. **If** $|R_{S-T}| > \eta + 3$ **then** A assumes B is connected to it through a wormhole

Fig. 2. The Initial Detection Algorithm

Select(R_{X-T})
1. $\forall X \in N_A - \{B\}$
   sort $|R_{X-T}|$ from longest to shortest
   ${R_{X-T}}^L$ is longest, ${R_{X-T}}^1$ is shortest
2. R_{S-T} = ${R_{X-T}}^L$
3. for$(i = 1,\ i < L,\ i++)$
   { **if**$\left(\left|{R_{x-T}}^L - {R_{x-T}}^{L-i}\right| \leq \eta\right)$
   then $R_{S-T} = {R_{X-T}}^{L-i}$
   else break }

Fig. 3. The Route Selection Algorithm

Selection of Route: The algorithm used by node A to find R_{S-T} is *Select*(R_{X-T}) and shown in Fig. 3. Node A creates a sorted list of route lengths from its neighbors to T (excluding replies from neighbors that do not have routes to the target node). Node A picks a route that is smaller than the longest route by *not more than* η if it exists. Otherwise the longest route is picked. It is the length of the picked route that is used to determine the existence of the wormhole. We have tested other methods for *Select*(R_{X-T}). Using the longest route has a better detection rate especially for short wormholes but increases the percentage of false positives for randomly distributed networks. Using the average length of all routes reduced false positives but also reduced the detection percentage. The method in Fig. 3 provides the best performance.

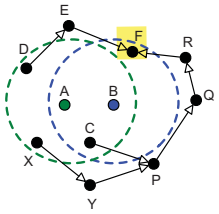

Fig. 4. Detection Operation without Wormhole

Selection of η: The value of η is not known a priori, but while implementing security in the network, the administrative entity can decide what it should be. Typically, longer the wormhole is, greater is the damage. With $\eta = 2$ even short wormholes (2 hops) can be detected. However, the simulation results will show that the number of false positives will be high. Using $\eta \geq 3$ reduces false positives but short wormholes (less than 3 hops) may escape detection. $\eta = 2$ or 3 provides the best tradeoff between detection rate and false positives.

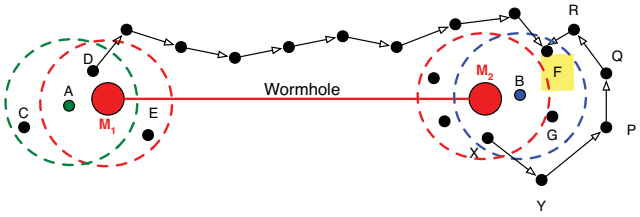

Fig. 5. Detection Operation with Wormhole - case 1

No Wormhole: In the example in Fig. 4 nodes A and B are real neighbors – there is no wormhole in this case. Node A wants to check if node B is a real neighbor. Node A picks node F as the target node and asks its neighbors C, D, and X to find routes to node F. The lengths of these routes will be 4, 2, and 5 hops. Note that the nodes have to avoid the one hop neighbors of nodes A and B in their routes to F. Node A will select one of these routes (for now let us assume it is the longest one of 5 hops). The route from X to F through A will be 3 hops. If $5 - 3 < \eta$ then node A will decide that node B is a real neighbor. In some cases, $|R_{X-T}| - 3 \geq \eta$ if the topology is sparse and there is a false positive.

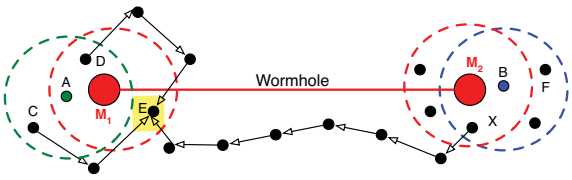

Fig. 6. Detection Operation with Wormhole - case 2

With Wormhole: Cases where nodes A and B are connected with a wormhole are shown in Figs. 5 and 6. In Fig. 5, $A \in N_{M_1}$ and it has at least one neighbor $B \in N_{M_2}$. The target

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

node must be a neighbor of B but not of A. Thus, there are two possibilities for the target node in this case. An example of the first is node $F \in \hat{N}_B - N_{M_2}$ in Fig. 5. Neighbors of A avoid other nodes in N_A and all nodes in N_B when they try to reach F. Since all nodes in N_{M_2} are included in N_A and all nodes in N_{M_1} are included in N_B, all the wormhole links will be avoided. True neighbors of A will have routes to F that are longer than 3 hops by at least η and the wormhole will be detected. For instance, node D, which is a true neighbor of node A cannot use nodes B or X to reach the target node F and will use the long route shown in Fig. 5. In the second case, the target node is an element of N_{M_1} but outside the range of A (e.g., node E in Fig. 6). In this case, true neighbors of A will find short routes to E, but purported neighbors $\in N_{M_2}$ will have long routes to E (e.g., node X is Fig. 6). To conclude, in either case, some neighbor of A will report a route whose length exceeds $3 + \eta$ and the wormhole is detected.

C. Improving the Detection Process

We present approaches to reduce false positives and to reduce the overall number of checks that must be performed between supposed neighbors in the network.

Reducing False Positives: False positives can occur in two ways - first when there is no wormhole and the topology is sparse resulting a long route to the target and second when node A tries to check for the existence of a wormhole between itself and a Type 3 neighbor (see Fig. 1). In the latter case, nodes $\in N_A^*$ will find long routes to a Type 3 neighbor. At this point, simply using the detection scheme will not enable distinguishing between Type 1 and Type 2 neighbors.

We have found that mutual detection (i.e., A checking if B is a true neighbor and B checking if A is a true neighbor) reduces the percentage of false positives significantly. A wormhole will be suspected to exist **if and only if** both A and B discover their links to be connected through a wormhole. When there is no wormhole, the target nodes for nodes A and B (see Fig. 4) when they perform mutual checks will be different. In most cases, even if A marks node B as connected through a wormhole, B will not or vice versa. Of course there will still be topologies where both A and B will still flag each other as connected through a wormhole, but this fraction is an order of magnitude smaller as shown in Section IV. Similarly, even if node A in Fig. 1 flags node C, node C will not flag node A as it has no neighbors $\in N_{M_2}$. Thus false positives are reduced and Type 3 neighbors correctly identified.

Reducing the Number of Checks: If every node checks each its neighbors to detect the presence of a wormhole, it results in a large overhead and delay, especially in dense networks with high average node degree. The question then is whether every node must check links with all of its neighbors or some nodes can be exempt from applying the detection process. Without any formal proof, we argue that the following are simple rules for checking for existence of wormholes that eliminate a large number of unnecessary checks as verified by simulations in Section IV.

1) If node A checks its link with node B and no wormhole is discovered, node B need not check its link with node A.
2) If node A checks its link with node B and a wormhole is discovered, node B must check its link with node A. This ensures that false positives are reduced as discussed previously.
3) If node A has checked its link to node B and vice versa, no wormhole is discovered, and any node C is a neighbor of *both* A and B, nodes A and B need not check their links with node C, and vice-versa.

More aggressive approaches may result in reducing the detection rates. For instance, we could exempt *all one hop neighbors* of nodes that did not detect a wormhole from checks. If the wormhole is only connecting exactly one node A with another node B located several hops away (see Fig. 7), both nodes A and B, may never check for the wormhole as one of their neighbors that is not within the wormhole range could have ruled out a wormhole. Similarly, it is not a good idea to exempt *all* other neighbors from checking links to a node if one of its neighbors did not detect a wormhole. For instance, in Fig. 7, if C did not detect a wormhole when it checks its link to A, which is true, then B may also not check its link to A even though there is a wormhole in between. By including the condition that B has to also be a neighbor of C, this possibility is averted.

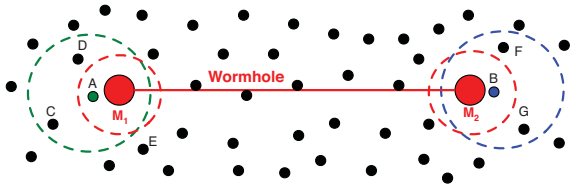

Fig. 7. Wormhole Connecting Single Nodes

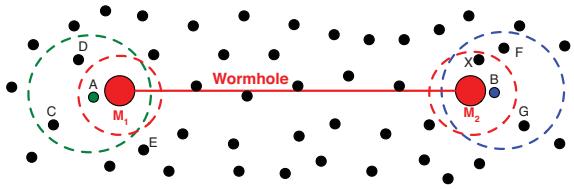

Fig. 8. Wormhole Connecting Single Node with Two nodes

However, one may expect the situation shown in Fig. 8 to be problematic, where node A has two neighbors X and $B \in N_A^*$. What if X checks its link with B and finds no wormhole? One of the characteristics of the detection process is that it cannot distinguish between Type 1 and Type 2 neighbors. Since both A and B are neighbors of X, when X checks its link with B, it will ask A to find a route to a target node (say G) and this will reveal the presence of the wormhole.

D. Removal of Bogus Links

Detecting the existence of wormholes in the network is an important step. However, another crucial process is to remove the links created by the wormhole. Note that a wormhole that connects m_1 nodes $\in N_{M_1}$ with m_2 nodes $\in N_{M_2}$ results in $2m_1m_2$ bogus links. Even if one of these links is not removed it will still cause damage by attracting traffic. Many

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

of the available wormhole defense mechanisms ignore the removal of the wormhole connected links or use techniques that may remove many legal links. As previously described, the detection process flags the existence of a wormhole. A link between both Type 1 neighbors and Type 2 neighbors will be flagged as corrupted by a wormhole (and this is confirmed by mutual checks). However, mutual checks between Type 3 neighbors allows them to identify the fact that they are not connected through a wormhole. The challenge then is to distinguish between Type 1 and Type 2 neighbors to avoid removing legal links between Type 2 neighbors.

Removal(A,B)

1. $\forall X \in N_A - N_B$,
   For some $Y \in N_B - N_A$,
   X **finds** R_{X-Y} : no element in $N_A \cup N_B$ is part of R_{X-Y}
   then X **sends** $|R_{X-Y}|$ to A
2. A uses all R_{X-Y}s and employs *Select*(R_{X-Y}) to **find** $|R_{S-Y}|$
3. **if** $|R_{S-Y}| > \eta + 3$
   then A removes link to B and the process will stop
   else all X will repeat process for next $Y \in N_B - N_A$
4. **if** no $|R_{S-Y}| > \eta \; \forall Y \in N_B - N_A$,
   then link A to B will not be removed

Fig. 9. Wormhole Removal Algorithm

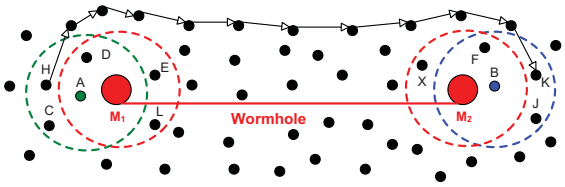

Fig. 10. Identification of Type 1 Neighbor

The removal algorithm that should be used by node A to decide the removal of the link to node B is shown in Fig. 9. If a node A *detects* the existence of a wormhole when it checks the link between itself and node B, all neighbors of A and B (i.e., all nodes in $N_A \cup N_B$) will use the algorithm in Fig. 9. The way this algorithm works is as follows. Node A will ask all its neighbors that are not part of N_B to find routes to neighbors in N_B that are not part of N_A *one-by-one*. If at any point, routes are found to be very long (similar to *Detection()*), the process stops, B is flagged as a Type 1 neighbor, and the link is removed. If not, node B is flagged as a Type 2 neighbor.

Why does this work? Consider a Type 1 neighbor (see Fig. 10) B of node A. Node A can have neighbors in $N_A - N_B$ that are on either side of the wormhole as also node B. This is the reason why nodes in $N_A - N_B$ should find routes to nodes in $N_B - N_A$ one-by-one. Eventually, a long route is discovered. For example, node $H \in \hat{N_A}$ will find a long route to $K \in \hat{N_B}$ or node $X \in N_A^*$ will find a long route to $E \in N_B^*$. It is not sufficient to pick any one node in $N_B - N_A$ as it is possible

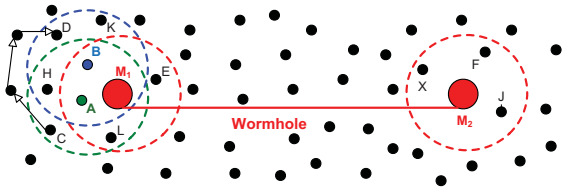

Fig. 11. Identification of Type 2 Neighbor

that $N_A - N_B$ has nodes on only one side or the other of the wormhole. In the case of a Type 2 neighbor B (see Fig. 11), nodes in $N_B - N_A$ and $N_A - N_B$ are *both* constrained to be on the same side as nodes A and B. Any neighbor of A that belongs to N_A^* will also belong to N_B^*. Thus, routes from nodes in $N_A - N_B$ to nodes in $N_B - N_A$ will likely be short (e.g., from node C to node D in Fig. 11).

A question that arises here is why do we not use this process for detection itself? The reason is that this results in a large number of false positives (i.e., when only nodes in $N_A - N_B$ find routes to nodes in $N_B - N_A$ one-by-one, it is likely that some outlying long route exists flagging it as a wormhole). We observed that the false positives can be as large as 35% compared to 1% using the *Detection()* process mutually between nodes. Using the $Removal()$ process mutually between two nodes A and B can reduce the number of legal links removed.

E. Other Issues

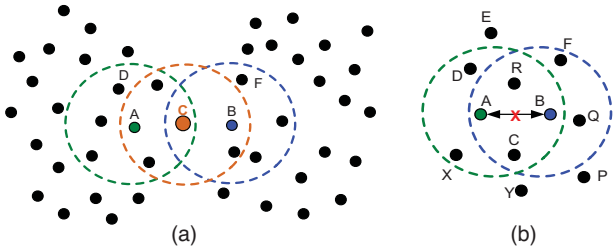

Fig. 12. (a) Critical Node (b) Removal of Legal Links

If a *critical node* (if this node is removed, the network will be partitioned) exists in the the network, then the $Detection()$ process will not work. Fig. 12(a) shows an example of a critical node in a network. Methods of addressing this problem are discussed in [3]. While the number of legal links removed will be small (as shown in Section IV), the impact of removing a very small number of legal links can be expected to be minimal. For example, in Fig. 12(b), if the link from A to B is removed, node A may be able to use node C to reach Q without an increase in the number of hops. In some cases, a few additional hops may be required. Finally, we have not explicitly described protocols that will use $Removal()$ (and employ mutual checks and exchange of information). We can expect this to operate in a manner similar to the detection process.

IV. Performance Evaluation

A. Simulations

The important performance metrics for secure neighbor creation are: the percentage of correct detection of the wormhole,

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

the percentage of false positives, percentage of wormhole links removed, and percentage of legal links removed by mistake. We have considered two different node distribution models: grid distribution with some perturbations and random distribution. For the random node distribution, the coordinates of the nodes $(x_i,\ y_i)$ for $i = 1, 2, ...200$ were independently and randomly chosen in the range from 100m to 2000m using a uniform [100-2000] random number generator. In the grid case, nodes are located in a perturbed 20×20 grid. The coordinates of each node x_i and y_j were randomly chosen using uniform random variables in the ranges $(100i - p100, 100i + p100)$ and $(100j - p100, 100j + p100)$, respectively, where p is the perturbation parameter and $i = 1, ...20$ and $j = 1, ...20$ (in our simulations, $p = 0.2$). As in [3], we also investigated SECUND with two connectivity models the commonly employed unit disk graph and the quasi unit disk graph. The quasi-UDG connectivity model is available in [22]. Only results with the UDG are shown here for brevity. The results with the quasi-UDG connectivity model are very similar. To change the average node degree, the transmission range of the nodes was varied from 110m to 160m. The simulations were programmed in C using DSR routing protocol and node distribution models from ns-2. For statistical validation the simulations were repeated 50 to 100 times with confidence intervals of 95%. SECUND was evaluated for networks without any wormhole, with two-ended wormholes, and multi-ended wormholes.

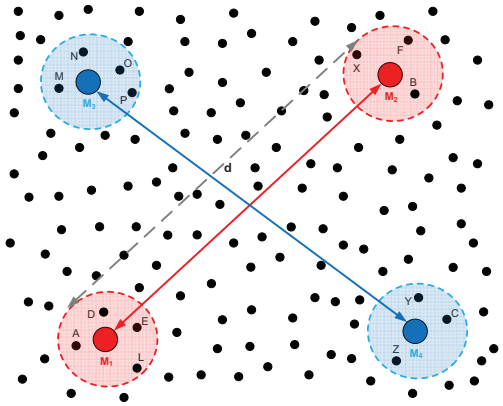

Fig. 13. Two-ended Regular Wormhole

Two-ended regular wormholes: The two-ended wormhole connects groups of nodes from one location in the network to another group of nodes located several hops away (wormholes with different lengths are tested). As shown in Fig. 13, two separate wormholes are created in the network such that the ranges of the wormhole transceivers do not overlap. That is, each node in M_1's (M_3's) range is only connected to every node in M_2's (M_4's) area and vice versa. Let m_i be the number of nodes in the range of wormhole transceiver M_i. The number of links created by wormhole $M_1 \leftrightarrow M_2$ is $2m_1m_2$. Note that two nodes $A \in N_{M_1}$ and $B \in N_{M_2}$ have two links between between them $A \rightarrow B$ and $B \rightarrow A$. For the two wormholes shown in Fig. 13, the number of bogus links created is $(2 \times 4 \times 3) + (2 \times 4 \times 3) = 48$.

Multi-Ended wormhole: In this case the wormhole will be connecting nodes located in many different areas. In the example shown in Fig. 14, each node located near any wormhole transceiver will be connected to all nodes located at the other transceivers. For instance, every node in N_{M_1} will be connected to every node in N_{M_2}, N_{M_3}, and N_{M_4}. In general, for a n-ended wormhole the number of links created will be: $\sum_{i=1}^{n-1}\left[m_i \cdot \left(\sum_{j=1, j\neq i}^{n-1} m_{j+1}\right)\right]$. For the example in Fig. 14 the number of links created by the 4-ended wormhole is $4\,(3+4+3)+3\,(4+4+3)+4\,(4+3+3)+3\,(4+3+4) = 146$.

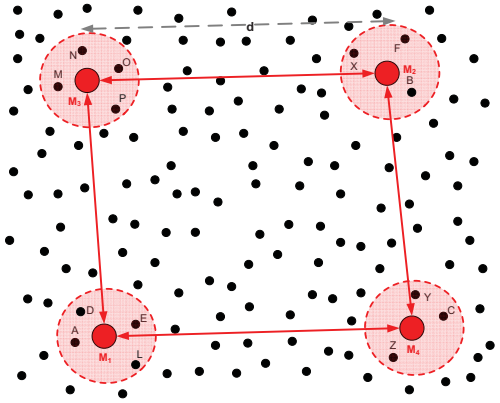

Fig. 14. Multi-End Wormhole

B. Results Without Wormholes

η	2	3	4	5
Grid	7.190	0.489	0.038	0.0172
Random	11.216	1.699	0.281	0.092

TABLE I
FALSE POSITIVES WITHOUT MUTUAL CHECKS

We simulate networks without wormholes and run the $Detection()$ and $Removal()$ algorithms to determine the percentage of false positives and legal links removed by mistake.

The false positive rate is determined by the fraction of instances where the $Detection()$ process flags a wormhole as existing when it does not. We first look at the false positive rate when *without* mutual checks (i.e., A runs $Detection()$ but B does not when A checks to see if B is a neighbor). Simulation results shown in Table I indicate that the false positive rate can be fairly high for small values of η. This is because it is possible for some nodes to only find routes to the target that are longer than the route through A and B by more than 2 hops. However, with $\eta = 3$ the percentage of false positives is less than 2%. When $Detection()$ is run by both nodes (mutual checks), the false positive rates fall drastically as seen in Table II. It is close to 0 for grid distributions of nodes and less than 0.2% for randomly distributed nodes for $\eta = 3$. More false positives occur with randomly distributed nodes since nodes may have relatively long routes to reach the target node.

The percentage of legal links removed by mistake (number of links removed/total number of links) when there is no wormhole, is a very important performance metric for secure neighbor discovery protocols. Table III shows the percentage

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

η	2	3	4	5
Grid	1.299	0.006	0	0
Random	2.884	0.196	0	0

TABLE II
PERCENTAGE OF FALSE POSITIVES WITH MUTUAL CHECKS

of legal links removed by mistake for both grid and randomly distributed nodes for different values of η. The results show that SECUND removes none or very few legal links. Note that in this case there are no Type 1 or Type 2 neighbors which makes the process less complicated. For example, with $\eta = 3$ no links are removed by mistake with the grid network and only 1 or 2 links are removed by mistake on average from the entire simulated network.

η	2	3	4	5
Grid	0.468	0.001	0	0
Random	1.033	0.061	0	0

TABLE III
PERCENTAGE OF LEGAL LINKS REMOVED BY MISTAKE

C. Results with Wormholes

In this section, we present simulation results when two-ended and multi-ended wormholes are present in networks with nodes distributed in a grid and randomly. We first present results of the percentage of legal links removed – in comparison with results from the previous section. We present detection rates as a function of wormhole length and node degree next.

1) Removal of Bogus and Legal Links: We simulated wormholes with $d \geq 5$ hops and considered the fraction of legal links removed and the fraction of bogus links removed for different values of η. In the case of both grid and random distribution of nodes and for both two-ended and multi-ended wormholes, $\eta = 3$ provides almost 100% removal of bogus links and removal of less than 1% of legal links. Even for $\eta = 2$, the performance can be considered to be very good. As η increases to 5, the fraction of bogus links removed drops to around 80% with random node distributions and multi-ended wormholes where it becomes more difficult to distinguish between Type 1 and Type 2 neighbors for larger η values. These results are shown in Figs. 15 and 16.

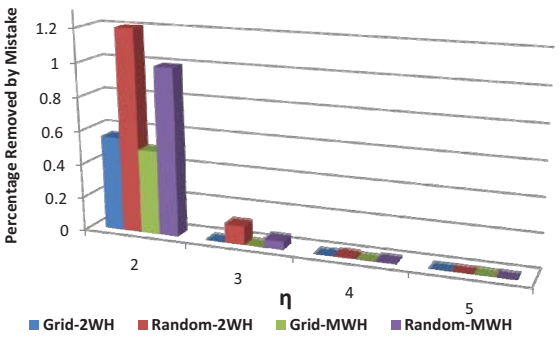

Fig. 15. Percentage of legal links removed with two and multi-ended wormholes

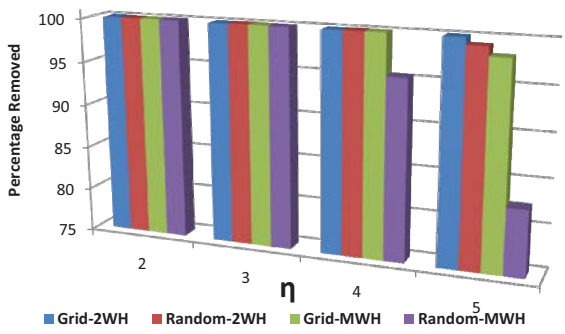

Fig. 16. Percentage of bogus links removed with two and multi-ended wormholes

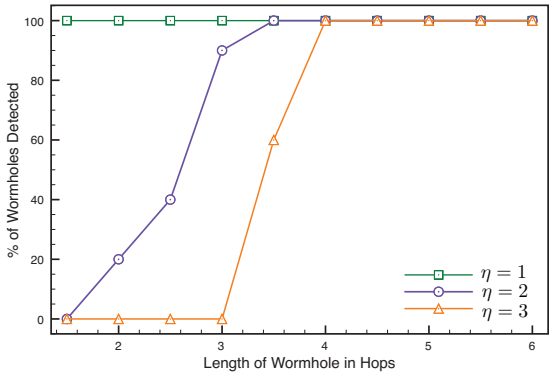

Fig. 17. Impact of wormhole length on Wormhole Detection

2) Impact of Wormhole Length: Figure 17 shows the percentage of wormholes detected for different values of wormhole length starting from 1.5 hops till 6 hops, and for $\eta = 1, 2, 3$. The results confirm that η impacts the length of the wormhole that can be detected. With $\eta = 1$, any wormhole can be detected but the number of false positives will be extremely high. With $\eta = 3$, any wormhole longer than 4 hops will be certainly detected. Similarly, the impact of the length of the wormhole on the removal process is shown in Fig. 18. The results show that the removal process will be enhanced with longer wormholes.

3) Impact of Node Degree: With $\eta = 3$ fixed, we simulated networks with a variety of average node degrees. The average node degree was changed by changing the transmission range of the nodes from 110 to 160 meters. Obviously, the larger the transmission range is, the more nodes there are that can be reached by a given node, and hence the higher the average node degree. A higher node degree provides more options for finding routes and improves the performance of SECUND in general. For example, the performance of $Detection()$ without mutual checks in terms of the % of false positives is shown in Fig. 19. With an average node degree of 5-6, the % of false positives is very small because it is unlikely that only outlying long routes to target nodes exist. Mutual checks further reduce the false positive percentages as shown in Fig. 20 where an average node degree of 4-6 has very low false positive rates. The impact of average node degree on the percentage of legal links removed by mistake for both grid and randomly

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

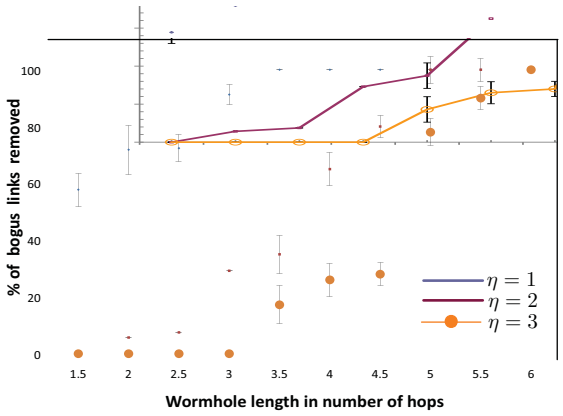

Fig. 18. Impact of wormhole length on Wormhole Removal

distributed networks is presented in Fig. 21. The trend in this case is similar to the trends with false positives.

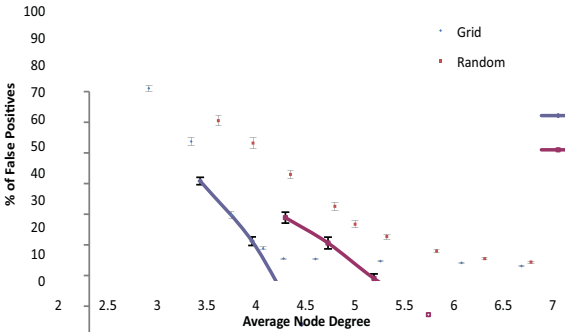

Fig. 19. False positives with detection without mutual checks

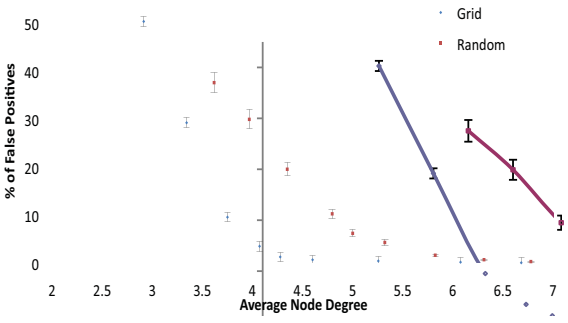

Fig. 20. False positives with detection with mutual checks

The impact of node degree on wormhole detection rates is shown in Fig. 22. For both grid and randomly distributed networks the results show that the detection process can detect wormholes successfully even for networks with very low node degree (3–3.5). Even lower node degrees (< 3) may result in nodes being unable to find alternate routes as required by SECUND. Similarly, an average node degree of 5 ensures that most bogus links are removed as shown in Fig. 23.

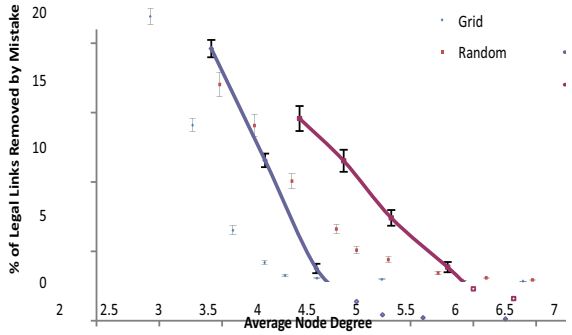

Fig. 21. Impact of Node Degree on Links Removed by Mistake

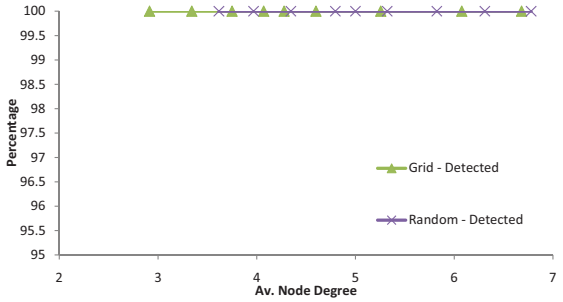

Fig. 22. Wormhole Detection Rates as a Function of Node Degree

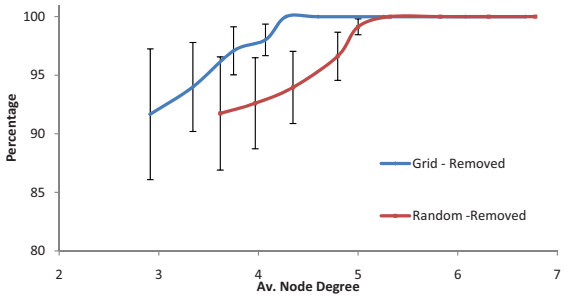

Fig. 23. Bogus Link Removal as a Function of Node Degree

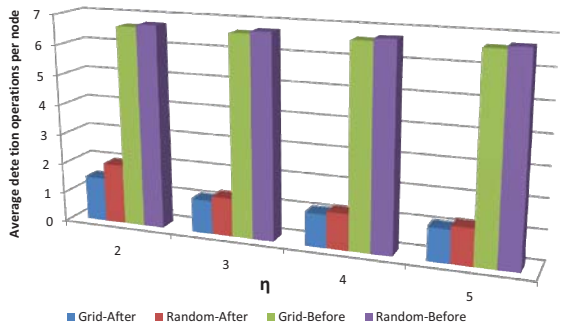

Fig. 24. Detection Operations Performed per Node

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380

4) Overhead Analysis: The number of $Detection()$ operations performed by each node (for a given network topology, density, and degree) impacts the average number of route acquisitions each node has to employ to create a secure neighborhood with SECUND. These statistics were captured in our simulations (the average node degree is 6.8).

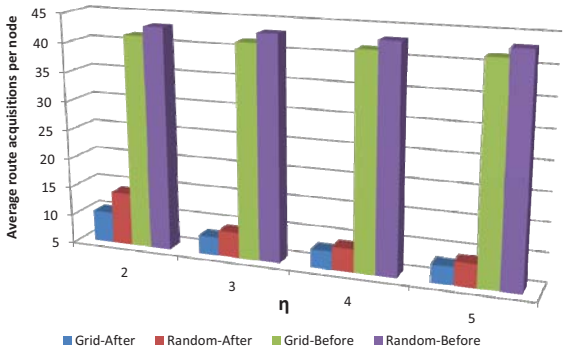

Fig. 25. Route acquisition per node

Fig. 24 shows the average number of times each node has to perform $Detection()$ before and after using the rules for cooperation in Section IIIC. With $\eta = 3$, instead of running $Detection()$ 6.8 times, each node runs it around two times after following the rules specified for improving the detection process. The number of route acquisitions that each node has to perform, shown in Fig. 25, falls from around 45 to 10, a savings in overhead of about 80%. Clearly, the rules presented in Section IIIC provide significant savings in overhead even though they are fairly simple without impacting performance metrics (and improving the performance as in the case of false positives).

V. Conclusions and Future Work

In this paper, we present a localized protocol SECUND, that employs cooperation between neighboring nodes for creating a secure neighborhood in ad hoc networks. SECUND employs routing hop count discrepancies between neighbors to determine the existence of a wormhole and to remove bogus links created by wormholes. SECUND incorporates simple rules for cooperation that result in its requiring only a small overhead in terms of number of links checked for wormholes. It does not have special requirements such as location information, very high node degree, accurate synchronization between nodes, special hardware etc. SECUND was tested through simulations for different distributions of nodes in networks and different types and lengths of wormholes. Under a variety of evaluated scenarios, SECUND demonstrates excellent detection probabilities and few false alarms, demonstrating its ability to detect wormholes in ad hoc networks. The protocol is also capable of removing most bogus links from the network while only removing very few legal links.

Acknowledgments

This work was funded in part by the Army Research Office MURI grant W911NF-07-1-0318.

References

[1] H. Yih-Chun, A. Perrig, and D. B. Johnson, "Wormhole attacks in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 370–380, 2006.

[2] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun, and J.-P. Hubaux, "Secure neighborhood discovery: a fundamental element for mobile ad hoc networking," in *Proc. of IEEE Communications Magazine*, 2008.

[3] T. Hayajneh, P. Krishnamurthy, and D. Tipper, "Deworm: A simple protocol to detect wormhole attacks in wireless ad hoc networks," in *In Proceedings of the IEEE Symposium on Network and System Security*, 2009.

[4] X. Wang and J. Wong, "An end-to-end detection of wormhole attack in wireless ad-hoc networks," in *In Proc. of International Conference on Computer Software and Applications*, 2007.

[5] Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *In Proce. of IEEE INFOCOM*, 2003.

[6] J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos, "Truelink: A practical countermeasure to the wormhole attack in wireless networks," in *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, S. V. Krishnamurthy, Ed., 2006, pp. 75–84.

[7] P. V. Tran, L. X. Hung, Y.-K. Lee, S. Lee, and H. Lee, "Ttm: An efficient mechanism to detect wormhole attacks in wireless ad-hoc networks," in *In Proc. of IEEE CCNC*, 2007.

[8] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Secure neighbor discovery in wireless networks: formal investigation of possibility," in *In Proceedings of the ACM symposium on Information, computer and communications security*, 2008.

[9] J. T. Chiang, J. J. Haas, Y.-C. Hu, P. R. Kumar, and J. Choi, "Fundamental limits on secure clock. synchronization and man-in-the-middle detection in fixed wireless networks," in *Proc. of IEEE INFOCOM*, 2009.

[10] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P. Hubaux, "A practical secure neighbor verification protocol for wireless sensor networks," in *Proc. of ACM WiSec*, 2009.

[11] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Network and Distributed System Security Symposium (NDSS)*, San Diego, 2004.

[12] S. Capkun, L. Buttya'n, and J.-P. Hubaux, "Sector: secure tracking of node encounters in multi-hop wireless networks," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*. Fairfax, Virginia: ACM, 2003, 986862 21-32.

[13] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of the 2nd ACM workshop on Wireless security*. San Diego, CA, USA: ACM, 2003, 941313 1-10.

[14] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proceedings of the 3rd ACM workshop on Wireless security*. Philadelphia, PA, USA: ACM, 2004, 1023657 51-60.

[15] R. Poovendran and L. Lazos, "A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks," *Wirel. Netw.*, vol. 13, no. 1, pp. 27–59, 2007, 1228615.

[16] L. Qian, N. Song, and X. Li, "Detection of wormhole attacks in multipath routed wireless ad hoc networks: a statistical analysis approach," *J. Netw. Comput. Appl.*, vol. 30, no. 1, pp. 308–330, 2007, 1238698.

[17] R. Maheshwari, J. Gao, and S. R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, J. Gao, Ed., 2007, pp. 107–115.

[18] Y.-T. Hou, C.-M. Chen, and B. Jeng, "Distributed detection of wormholes and critical links in wireless sensor networks," in *Proc. of IIHMSP*, 2007.

[19] C. Lee and J. Suzuki, "Swat: A decentralized self-healing mechanism for wormhole attacks in wireless sensor networks," *In Y. Xiao, H. Chen and F. Li (eds.) Handbook on Sensor Networks, Chapter 24, World Scientific Publishing, ISBN: 978-981-283-730-1*, 2010.

[20] W. Znaidi, M. Minier, and J.-P. Babau, "Detecting wormhole attacks in wireless networks using local neighborhood information," in *Proc. of IEEE PIMRC*, 2008.

[21] I. Khalil, S. Bagchi, and N. B. Shroff, "Liteworp: Detection and isolation of the wormhole attack in static multihop wireless networks," *Comput. Netw.*, vol. 51, no. 13, pp. 3750–3772, 2007, 1276793.

[22] F. Kuhn and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," in *Proceedings of the 2003 joint workshop on Foundations of mobile computing*. San Diego, CA, USA: ACM, 2003, 941089 69-78.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8380
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8380