A Practical Secure Neighbor Verification Protocol for Wireless Sensor Networks

Reza Shokri, Marcin Poturalski,
Gael Ravot, Panos Papadimitratos, and Jean-Pierre Hubaux
Laboratory for Computer Communications and Applications, EPFL, Switzerland
firstname.lastname@epfl.ch

ABSTRACT

Wireless networking relies on a fundamental building block, *neighbor discovery* (ND). The nature of wireless communications, however, makes attacks against ND easy: An adversary can simply replay or relay (wormhole) packets across the network and mislead disconnected nodes into believing that they communicate directly. Such attacks can compromise the overlying protocols and applications. Proposed methods in the literature seek to secure ND, allowing nodes to *verify* they are neighbors. However, they either rely on specialized hardware or infrastructure, or offer limited security. In this paper, we address these problems, designing a practical and secure *neighbor verification* protocol for constrained Wireless Sensor networks (WSNs). Our scheme relies on estimated distance between nodes and simple geometric tests, and it is fully distributed. We prove our protocol is secure against the classic 2-end wormhole attack. Moreover, we provide a proof-of-concept implementation with off-the-shelf WSN equipment: Cricket motes.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design —*Wireless communication*

General Terms

Security, Design, Experimentation

1. INTRODUCTION

Neighbor discovery (ND) provides an essential functionality for wireless devices, that is to discover other devices that they can communicate directly through the wireless medium. This is a fundamental building block for wireless networking, routing being the most essential in the context of Wireless Sensor Networks (WSNs). Nonetheless, the nature of wireless communications makes it easy to abuse ND. Consider two sensor nodes A and B, out of each other's communication range, and an adversary that controls two relay nodes M and W, within range of A and B, respectively. The adversarial node M receives packets from A, relays them to W, and then retransmits them to B. The result is that, through a false link, A and B are misled to believe that they are neighbors, although they are not. Consequently, such an adversary can control the victims' communication, or more generally, he can control multi-hop communication by shortening communication paths and manipulating other system's operations.

The importance of "wormhole" or "relaying" attacks in wireless networks has been identified in the literature. The importance of these attacks is due to the fact that it is easy for an adversary to act at the physical layer without any node compromise or possession of cryptographic keys. *Secure neighbor verification* thwarts this attack, essentially, by identifying as neighbors only those nodes that are indeed neighbors. However, a neighbor verification protocol does not have to be complete, i.e. to discover and verify all actual neighbors. This is, most notably, due to the jamming attack that the adversary can always perform to prevent the discovery of legitimate neighbors.

As corroborated by a recent impossibility result for a broad class of secure wireless ND protocols (run by two correct nodes in the presence of the wormhole), it is quite challenging, in general, to verify *communication* neighborhood, i.e., to discover other nodes that can directly communicate with a considered node [12]. Hence, in practice, secure ND protocols often approximate it with *physical* neighborhood, where nodes prove they are in vicinity of each other.

A number of solutions to secure ND have been proposed thus far. However, as we show in Section 2, virtually all of them are not generally applicable to WSN, either because they need specialized hardware, or they rely on the presence of an external trustworthy infrastructure, or because they are based on assumptions that limit their applicability.

In this paper, we design a powerful and secure neighbor verification protocol that adheres to the limited hardware capabilities of WSN, as it is demonstrated by our implementation. In our protocol, each node estimates its distance to the other nodes it can communicate with through a single hop. Then, nodes exchange information about their estimates. Next, a series of simple geometric tests is run by each node over the local neighborhood view it has obtained, in order to detect topology distortions created by wormhole attacks. Only those nodes that successfully pass the tests are verified to be actual communication neighbors.

Our results show that these tests are highly effective. To create false links, the adversary needs to carefully adjust the distance between victims by selecting the appropriate delays introduced when relaying messages. The adjustment is based on the location of the motes. Without centimeter-precision knowledge of location of victim nodes, which is hard to obtain, the adversary has only a negligible chance to create false links. Even with that knowledge, we show that the adversary cannot create any false links with a 2-end attack. We also show, using simulations, that the number of false links that an adversary can create with a k-end wormhole is highly limited. The latter results are omitted from this version of the paper due to space limitations, and we refer the reader to the associated technical report [15].

Our solution is independent of the technology that wireless devices use to estimate their distance to each other. Nonetheless, our proof-of-concept implementation relies on off-the-shelf WSN equipment: Cricket motes that are capable of ultra-sound (US) ranging. However, the simple ranging protocol that is implemented in Cricket motes is replaced with our secure ranging protocol, which is tailored to our security requirements. Overall, our contributions in this paper are: (i) a secure ND scheme tailored for WSN, (ii) the provable security[1] of the protocol against a 2-end wormhole, (iii) the implementation and experimental evaluation of the protocol with existing WSN hardware.

2. RELATED WORK

A number of secure neighbor discovery schemes have been proposed in the literature. We briefly survey schemes that are not generally applicable to WSN, because of their special hardware requirements or dependance on additional infrastructure. Then, we discuss in details the applicable schemes and compare them with our proposal.

The scheme proposed in [4] detects wormhole attacks using directional antennas, that are not common for WSN. RF-fingerprinting, i.e., recognizing a wireless transceiver based on unique features of the signals it generates, is considered in [13]. However, the signal analysis that this scheme performs is not feasible for a typical sensor mote radio receiver. Geographical packet leashes [5] rely on location information to estimate the distance between the nodes, and thus prevent wormhole attacks. However, node knowledge of *secure* location is hard to obtain in many settings. In particular, secure localization schemes for WSN require additional infrastructure such as a number of location-aware beacons [16]. This type of additional infrastructure is also the basis of a wormhole prevention scheme proposed in [11].

Another method to estimate the distance between wireless nodes uses the message time-of-flight (through the US [14], or RF medium [5][17]) between two nodes. US-based distance measurement is not secure in general: An adversary can decrease the distance by relaying a slow US signal over a fast relay link. The RF-based method is also problematic, especially in the context of sensor networks, given the light-speed propagation of RF waves. To precisely estimate the distance, a nanosecond precision clock is required, that is well beyond the capabilities of existing sensor motes.

Nevertheless, it is possible to measure the RF time-of-flight with the best precision available, as proposed in [9] for generic wireless networks, in [3] for 802.11, or [1] for WSN. Actually, our scheme utilizes this method (in the *synchronization* phase). The microsecond precision clock available on sensor motes allows us to thwart store-and-forward wormhole attacks, and to prevent the adversary from creating very long false links. However, false links spanning over few hops are not detected. Hence, our protocol also includes more sophisticated defense mechanisms.

Two centralized approaches that rely on distance measurements [18] or only connectivity information [2] are proposed. Both are essentially implementable on WSNs, but have some drawbacks. The former visualizes the network and requires user interaction to localize the wormholes, limiting its applicability. The latter detects wormhole attacks based on the abnormal values of some statistics of the connectivity graph, but this requires assumptions about the expected values of these statistics. Moreover, no information about the location of the detected wormhole is provided.

A distributed scheme that relies only on connectivity information detects wormhole attacks by checking for forbidden structures in the connectivity graph [7]. A forbidden structure is a graph that is very unlikely to be observed as a subgraph of a legitimate connectivity graph, but that often appears under the attack: The precise form of these structures depends on the connectivity pattern. This method demonstrates good performance (specifically high detection rates) for dense networks when evaluated (with simulations) under theoretical connectivity patterns (unit-disk (UDG), quasi-UDG and the TOSSIM models). However, the scheme is not investigated in a real deployment, with a high irregularity of antenna patterns [20]. Under such conditions, the authors of [7] propose to empirically estimate the forbidden structures in an attacker-free part of the network. However, it might be impossible, or prohibitively expensive, to verify that a part of the network is wormhole-free. Besides, a part of the network does not necessary represent the entire network accurately. Furthermore, the security of this scheme is not analyzed under k-end relay attacks.

3. SYSTEM MODEL

We assume a static WSN where nodes (*motes*), A, B, ..., are distributed throughout a field. For the sake of simplicity, our analysis assumes nodes are deployed on a plane, but our protocol can easily be extended for nodes deployed in three dimensions. Nodes are *correct*, i.e., follow the protocol. The distance from A to B as measured by A, is denoted by d_{AB}, whereas $|AB|$ denotes the actual (Euclidean) distance between nodes (or points) A and B.

Each node is equipped with a microsecond precision clock, and two network interfaces: a radio-frequency (RF) and a sound (typically, ultra-sound (US)) interface. R is the range of the US technology, $s = 342$m/s is the speed of sound, and $c = 3 \times 10^8$m/s is the speed of light. We call two nodes, A and B, neighbors if and only if they are in physical proximity of each other, i.e., $|AB| < R$, and they can communicate directly in a symmetric way using both US and RF.

Nodes can perform cryptographic operations with a symmetric key K: encryption ($E_K\{.\}$), message authentication code ($MAC_K\{.\}$), and hash or one-way functions ($H\{.\}$) computations. They can also generate fresh random nonces. Every pair of nodes, A, B, running a ND protocol shares a symmetric key, K_{AB}.

[1]Except an arguably infeasible case where the adversary has to mount its relay node on top of a victim.

Adversary Model We are only concerned with *external* adversaries that cannot compromise correct nodes or their cryptographic keys. The adversary controls a number of *relay nodes* we also term as *wormhole ends.* The relay nodes are connected through an out-of-band *relay link*, over which information propagates with speed equal to the speed of light, c. Every relay node is equipped with one radio (RF) interface, and exactly one ultra-sound (US) interface with an omnidirectional transceiver.

We assume the adversary is sophisticated enough to relay messages on a *per-symbol* basis (i.e. messages are relayed at the physical layer and the adversary is not limited to *store-and-forward* relaying). The processing time at wormhole ends is considered to be below 1μ second. As the reception time of an US message is estimated with microsecond precision, such relaying delay has a negligible effect on US-based distance measurements. This is so when the attacker relays US signals over short relay links. However, long relay links (above 300m) introduce a non-negligible delay (a few microseconds) to relayed messages.

Messages (both RF and US) eavesdropped by a relay node can be selectively discarded, modified, delayed, or replicated before being relayed by another node. The adversary can also inject new messages into the network, but it is computationally bounded and unable to mount cryptanalytic attacks or to guess fresh nonces.

4. THE SCHEME

In the initialization phase of the network, when nodes discover their neighbors, there is no guarantee that the nodes that appear to be neighbors are indeed so. Our secure neighbor verification protocol will address this, relying on the previously established *security associations* (SAs) using a key establishment protocol (e.g.,[21]). The protocol is intended to run after most of the WSN nodes are deployed and SAs are established. If the network topology changes significantly, as the operator relocates nodes or deploys new ones, the protocol is re-run. The protocol consists of the following phases:

(i) **Ranging** Given a list of potential neighbors, every node calculates its distance to all of its neighbors using our US-based ranging protocol.

(ii) **Neighbor Table Exchange** After the ranging phase, every node shares with each of its neighbors (in an *authenticated* manner) the neighbor table, including the distances calculated in the ranging phase. Then, each node creates a table that includes the distances between its neighbors in addition to its own distance to its neighbors.

(iii) **Link Verification** The extended neighbor table is examined by a number of security tests. As a result, each link is *verified* in the case it passes specific tests. Otherwise, it is marked as an *unverifiable* link.

The protocol operates with a set of parameters adjusted based on the physical characteristics of the communication channel. ϵ_{sym} is the upper bound on the amount of link asymmetry error. ϵ_{quad} is the upper bound on the error of determining if four points are part of a quadrilateral on a plane (e.g., being a tetrahedron). Finally, ϵ_{sync} is the bound on propagation delay estimation error.

4.1 Ranging

A three-step *ranging protocol* is run once by every node in the network. The first step of the ranging phase, that is performed over RF, allows the *initiator* node, A, to *synchronize* itself with its neighbors (with microsecond precision). In this step, nodes do not adjust their clocks after synchronization; rather, the difference between clock readings at A and at a neighbor B is determined, which will be used in security tests and distance measurement. The second step, performed over US, is the actual *ranging*. This operation is done *simultaneously* for all neighbors, by broadcasting an ultra-sound message. The last step, *acknowledgement*, finalizes the ranging and secures the synchronization. Figure 1 illustrates one execution of the ranging protocol between node A and two of its neighbors B and C. Next, we focus on explaining communication between A and B in detail.

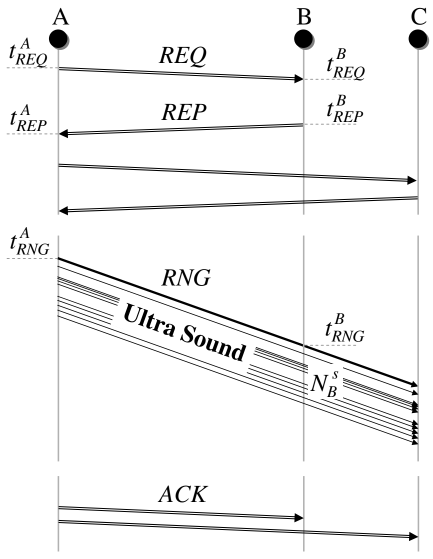

Figure 1: Ranging Protocol. The *initiator* node A is ranging its neighbors B and C.

Synchronization The initiator, A, sends via RF a REQ message to B, and it records the time of transmission as t^A_{REQ}. The REQ message contains an encrypted freshly generated nonce, N^r_B, which is different for every neighbor B and is used to authenticate B's response. It also contains the hash function value of another fresh nonce N^s, used in the ranging phase, which is the same for all neighbors of A. The whole message, including the header, is authenticated with a MAC using the shared secret key between A and B. Having received such a message, if the MAC is correct and N^r_B has not been seen or generated as a nonce before, node B records the time of reception t^B_{REQ} and replies with the REP message, which contains the nonce, N^r_B, while recording the time of sending as t^B_{REP}. When A receives the REP message, it also records the time of reception, t^A_{REP}. This procedure is repeated for each neighbor of A.

$$\begin{array}{lll}
A \xrightarrow{\text{RF}} B: & \langle REQ, E_{K_{AB}}\{N^r_B\}, H(N^s), MAC_{K_{AB}}\{.\}\rangle \\
A: & t^A_{REQ} := \text{Sending time of } REQ. \\
B: & t^B_{REQ} := \text{Reception time of } REQ. \\
B: & \text{If } N^r_B \text{ is fresh and MAC is correct, then:} \\
B \xrightarrow{\text{RF}} A: & \langle REP, N^r_B\rangle \\
B: & t^B_{REP} := \text{Sending time of } REP. \\
A: & t^A_{REP} := \text{Reception time of } REP.
\end{array}$$

Ranging The initiator, A, broadcasts the ranging message RNG over US and records the time of transmission as t^A_{RNG}. The message consists of a single bit 1, as the preamble, concatenated with the nonce N^s generated in the synchronization step. Every neighbor B, upon receiving this message, records the time as t^B_{RNG} and the subsequent bits (which is the nonce) as N^s_B.

$A \xrightarrow{\text{US}} * :$ $\langle 1 \| N^s \rangle$
$A :$ $t^A_{RNG} :=$ Sending time of RNG
$B :$ $t^B_{RNG} :=$ Reception time of RNG
$B :$ $N^s_B :=$ Received RNG nonce

Acknowledgement To conclude the ranging operation, the initiator, A, sends an ACK message to those neighbors *that have replied correctly in the synchronization step.*

$A :$ If B has sent correct REP, then:
$A \xrightarrow{\text{RF}} B :$ $\langle ACK, N^s, t^A_{REQ}, t^A_{REP}, t^A_{RNG}, MAC_{K_{AB}}\{.\}\rangle$
$B :$ If MAC is correct and $N^s = N^s_B$, and $|(t^A_{REP} - t^A_{REQ}) - (t^B_{REP} - t^B_{REQ})| < \epsilon_{sync}$:
$d_{BA} := ((t^B_{RNG} - t^B_{REQ}) - (t^A_{RNG} - t^A_{REQ})) \times s$

The ACK message contains N^s, to bind it to the earlier transmitted REQ and RNG messages, and three time-stamps, t^A_{REQ}, t^A_{REP} and t^A_{RNG}, that allow B to assure if the synchronization was done correctly, i.e., no delay beyond the propagation delay (which is below $1\mu s$ for RF communication at the range we are considering) was introduced while the messages were in transit. Node B also checks if the received nonce in the ranging phase, N^s_B, is equal to N^s (that was sent by A). If the checks are successful, B uses the time-stamps to calculate its distance to A. To provide authentication and integrity, the ACK message includes also the MAC over all its content.

4.2 Neighbor Table Exchange

The neighbor table NT_A, constructed by every node A in the network, contains the *identifiers* of its neighbors that were previously properly ranged and its *distance* to them. Neighbors for which ranging failed are not included in the table. The table is broadcasted (authenticated for all neighbors) within the NTE message, as shown below.

$$A \xrightarrow{\text{RF}} * : \quad \langle NTE, NT_A, MAC_{K_{AB}}\{.\}, MAC_{K_{AC}}\{.\}, \ldots \rangle$$

Every node A combines its own NT_A and the received neighbor tables from its neighbors into a new neighbor table, $NT2_A$, to be used in the link verification phase. $NT2_A$ is the set that is composed of A itself and its neighbors, plus the distances between members of this set.

4.3 Link Verification

The link verification consists of the following consistency tests that node A runs on $NT2_A$. If at any point a link is discarded, it is ignored in the subsequent steps.

(i) **Link Symmetry Test**: Any link (U, V) is discarded if d_{UV} is different from d_{VU}. This includes links for which only one measurement is available.

```
for all $U, V \in NT2_A$ do
  if $|d_{UV} - d_{VU}| > \epsilon_{sym}$ then
    remove links $(U, V)$ and $(V, U)$ from $NT2_A$
```

(ii) **Maximum Range Test**: Links longer than the range R are discarded.

```
for all $U, V \in NT2_A$ do
  if $d_{UV} > R$ then
    remove links $(U, V)$ and $(V, U)$ from $NT2_A$
```

(iii) **Quadrilateral Test**: For every neighbor B [2], node A looks for two other nodes C and D that together they form a 4-clique (i.e., all four are neighbors of each other, and A knows the distances between all of them). The links to B, C, and D are declared *verified* if four of the nodes form a quadrilateral that is *convex*. We note that it is possible that after all the 4-cliques that include (A, B) as a link have been checked, the link is not verified. In this case, the link is declared as *unverifiable*. How to treat unverifiable links depends on the security policy of the network.

```
for all $B \in NT2_A$ s.t. link $(A, B)$ is not verified do
  for all $C, D \in NT2_A$ s.t. $\{A, B, C, D\}$ is a 4-clique and
  $\forall U, V \in \{A, B, C, D\}$, link $(U, V)$ is not unverifiable do
    if $\{A, B, C, D\}$ is a convex quadrilateral then
      mark all links in $\{A, B, C, D\}$ verified
  if link $(A, B)$ is not verified then
    mark it as unverifiable
```

To test if a 4-clique forms a quadrilateral, we check if it is possible to assign (locally) positions to the nodes such that no contradiction arises. This can be done as follows: We assign position $(0, 0)$ to A, and $(0, d_{AB})$ to B; then we check if d_{AB}, d_{BC}, d_{CA} form a triangle (i.e., if all three triangle inequalities hold). If not, there is a contradiction. Otherwise, we assign position (x_C, y_C) to C, where $y_C = (d^2_{AB} + d^2_{AC} - d^2_{BC})/(2d_{AB})$ and $x_C = (d^2_{AC} - y^2_C)^{\frac{1}{2}}$. Further, we check triangle inequalities for d_{AB}, d_{BD}, d_{DA}; if they hold, there are two possible positions for D: (x_D, y_D) and $(-x_D, y_D)$, where $y_D = (d^2_{AB} + d^2_{AD} - d^2_{BD})/(2d_{AB})$ and $x_D = (d^2_{AD} - y^2_D)^{\frac{1}{2}}$. For both positions, we check if the difference between the measured length d_{CD} and the length of CD determined by the calculated positions of C and D is within ϵ_{quad} bounds. If it is not in both cases, the distances do not belong to any quadrilateral. Otherwise, we assign the matching position to D. Here, ϵ_{quad} is computed based on the empirical distance measurement error.

Given the positions computed in the quadrilateral test above, to test if the quadrilateral is convex we compute the product of four cross products $(\overrightarrow{AB} \times \overrightarrow{BC})(\overrightarrow{BC} \times \overrightarrow{CD})(\overrightarrow{CD} \times \overrightarrow{DA})(\overrightarrow{DA} \times \overrightarrow{AB})$. If the product is positive, the quadrilateral is convex, whereas if it is negative the quadrilateral is concave. We ignore the quadrilateral if any three nodes are co-linear. This is tested by checking if the absolute value of any of the above cross products is lower than ϵ_{quad}.

5. SCHEME ANALYSIS

We first explain and prove the properties of our ranging protocol in Sections 5.1 and 5.2. We note that the ranging protocol itself mitigates a large class of attacks, as demonstrated by properties (r2) and (r3). Then, in Section 5.3 we prove that our scheme is secure against 2-end wormhole attacks. Finally, we evaluate its performance in a benign setting in Section 5.4. For the case of k-end wormhole attacks, we refer the reader to the corresponding technical report [15].

[2]The computational complexity of the protocol decreases if A iterates over its neighbors sorted in the ascending order, based on their connectivity degree in $NT2_A$.

5.1 Ranging

For this part of the analysis, we assume A and B to be correct nodes executing the ranging protocol, with A as the initiator. We will show that the ranging protocol satisfies the following properties:

(r1) If the protocol terminates successfully and B calculates d_{BA}, then either $d_{BA} \approx |AB|$ or $d_{BA} \approx (|AW_A| + |W_B B| + \tau)$, where W_A and W_B are the relay nodes close to A and B, respectively, and τ is the distance corresponding to the additional delay that the adversary might impose on the RNG message).

(r2) The protocol prevents the creation of very long false links, i.e., links that span more than $\frac{c}{2} \times \epsilon_{sync}$.

(r3) The protocol prevents the creation of any false links by a store-and-forward adversary.

According to the ranging protocol, if B calculates d_{BA}, the following events have happened (in this order, t_i are in global time):

(1) At time t_1, recorded as t^B_{REQ}, node B has received message $m_1 = \langle REQ, E_{K_{AB}}\{N^r_B\}, H(N^s), MAC_{K_{AB}}\{.\}\rangle$, with nonce N^r_B not seen or generated before by B;

(2) At time t_2, recorded as t^B_{REP}, node B has sent message $m_2 = \langle REP, N^r_B\rangle$;

(3) At time t_3, recorded as t^B_{RNG}, node B has received a RNG message $m_3 = 1||N^s$, with N^s neither seen nor generated before by B;

(4) Node B has received the following message with t^A_{REQ} and t^A_{REP} passing the synchronization test:
$m_4 = \langle ACK, N^s, t^A_{REQ}, t^A_{REP}, t^A_{RNG}, MAC_{K_{AB}}\{.\}\rangle$

Message m_1 is authenticated, and hence can be generated only by A or B. As B checks that N^r_B is not self-generated, node A had to generate m_1 and send it at some point in time $t_0 < t_1$, recorded as t^A_{REQ} by A.

Likewise, m_4 can be generated only by A: Node B is ruled out because it would never generate an ACK message for a nonce N^s it has not generated itself. Node A would only send the ACK message for B, if at some time $t_{2.1}$, recorded as t^A_{REP}, it had received a REP corresponding to its REQ message that was sent at t_0. The only acceptable REP message is m_2; as the nonce N^r_B is fresh and it was encrypted in m_1, only B could have generated this REP message. Hence, $t_2 < t_{2.1}$ and therefore $t_0 < t_1 < t_2 < t_{2.1}$. Thus, we conclude that within an acceptable bound ϵ_{sync} at time t_1 the clocks of A and B are equal to t^A_{REQ} and t^B_{REQ}, respectively, if the synchronization test is successful in (4).

From the above, we instantly get (r2): The round trip propagation delay is equal to $(t_{2.1} - t_0) - (t_2 - t_1) < \epsilon_{sync}$, which means that if the distance between A and B is longer than $\frac{c}{2} \times \epsilon_{sync}$, the ranging will fail and (A, B) will be discarded. Hence, the attack of a store-and-forward adversary, which cannot relay a RF message before it entirely receives it, is aborted in the synchronization phase. This is because such an attacker introduces an additional delay, equal to the duration of the exchanged messages, on the synchronization phase. This delay is well above ϵ_{sync}, hence (r3) follows.

When A broadcasts the RNG message m_3 at some time $t_{2.2}$, it is the first time N^s is sent in clear. Thus, it is not possible for a relay node, W_B, to send m_3 before another relay node, W_A, receives m_3 sent by A at $t_{2.2}$. Moreover, the integrity of m_4 is protected, hence B learns t^A_{RNG} that is the actual time of sending m_3 in A's clock. As we have shown that the synchronization phase is secure, B measures correctly, within an acceptable bound, the difference between $t_{2.2}$ and t_3, the time at which it received the nonce. Thus, the distance to A calculated by B is either $|AB|$, or $|AW_A| + |W_B B| + \tau$. Hence, (r1) follows.

5.2 Symmetry Test

We show that the delay that is introduced on a relay link must be the same in both directions for all false links the attacker creates over that relay link. Otherwise, the attacker cannot successfully pass the symmetry test for some of the links over the wormhole. Assume node A is on one side of the relay link, and nodes B and C are on the other side. The adversary relays the RNG message, sent by node B and received (over the wormhole) by node A, with delay τ_1, and relays the RNG message sent by node C, also received (over the wormhole) by node A, with delay τ_2. Node A also sends a RNG message, which needs to be received by both B and C if links (A, B) and (A, C) are not to be discarded by the link symmetry test. To further satisfy the symmetry test the adversary needs to relay the RNG message of A with delay τ_1 (for symmetry of link (A, B)) and at the same time with delay τ_2 (for symmetry of link (A, C)). Hence, $\tau_1 = \tau_2$.

5.3 Security against 2-end Wormholes

In this section, we investigate the security of our scheme against the most commonly considered relay attack: the *single wormhole*, that is a relay attack utilizing two remote wormhole ends. We assume the wormhole to be *remote*, in other words, no valid link exists between correct nodes on opposite sides of the wormhole (i.e., in vicinity of relay nodes). Moreover, we limit the analysis to the case where there is no obstacle in the network.

It is easy to show that an attempt to relay a RNG message between two nodes A and B on the same side of the wormhole fails. Indeed, the RNG message sent by A will reach B directly before the relayed one does. The latter message will either be ignored, or it will collide with the former. In neither case the relaying leads to an incorrect distance calculation. Therefore, the adversary can only effectively relay over the wormhole.

For convenience, we denote the areas on the opposite sides of the wormhole as the left side of the wormhole and the right side. To create a false link, the adversary needs to convince 4 nodes that they form a convex quadrilateral. There are two choices in terms of the number of victim nodes located on both sides of the wormhole: 2 nodes on each side (**2-2**), and 3 nodes on one and 1 node on the other side of the wormhole (**3-1**).

(**2-2**): We show that for this case the adversary is unable to create any false links.

Denote the nodes on the left side of the wormhole as A, B and the nodes on the right side of the wormhole as C, D. Links AB and CD need to be legitimate, whereas links AC, AD, BC and BD will be created over the wormhole (see Figure 2(a) for notation). There are two ways in which a convex quadrilateral can be formed:

1. AB and CD are the opposite sides of a quadrilateral (Figure 2(b)). In this case:

$$|AC| + |BD| = a + b + c + d + 2t = |AD| + |BC|$$

However, in a quadrilateral (of positive volume) the sum of diagonals' length (|AD| + |BC|) is always greater than the sum of two opposite sides (|AC| + |BD|). This contradiction implies that the adversary cannot succeed with this construction.

2. AB and CD are the diagonals of a *convex* quadrilateral (Figure 2(c)). Denote X to be the intersection of AB and CD, with $|AX| + |XB| = x$ and $|CX| + |XD| = y$. The triangle inequalities for triangles ACX, BCX, BDX, DAX in Figure 2(c), added together, give $x + y > a+b+c+d+2t > a+b+c+d$. However, the triangle inequalities for ABW_1 and CDW_2 in Figure 2(a) give $x + y < a + b + c + d$. This (second) contradiction implies the adversary cannot create a false link in the 2-2 case.

(**3-1**): For this case, depicted in figure 2(d), the adversary cannot create any false links, unless $t + d = 0$.

First, assume that $\tau = t + d > 0$, and also the adversary can convince the correct nodes that $ABCD$ is a quadrilateral. This means that AD, BD and CD intersect in a single point (Figure 2(e)). Then $|AD| - |AW_1| = \tau$, $|BD| - |BW_1| = \tau$ and $|CD| - |CW_1| = \tau$. This implies that A, B and C lie on a hyperbola with foci W_1 and D, more precisely on the hyperbola arm closer to W_1 (Figure 2(f)). As this hyperbola arm is concave, 3 points A, B, C cannot form a convex quadrilateral with point D and hence they cannot validate false links AD, BD or CD. Therefore, no false links can be verified.

On the contrary, if $t + d = 0$, from the perspective of correct nodes $D = W_1$, and $ABCD$ forms a convex quadrilateral. Hence all the false links AD, BD and CD will be verified. However, the applicability and effectiveness of this attacks are limited. For the equality to hold, the adversary must add negligible delay when relaying the message, and the wormhole end W_2 needs to be located in the same place as node D. Deploying a relay node at a distance less than a few centimeters from a sensor mote might be difficult considering their physical appearance. Besides, such a closely placed relay node would act as an obstacle and would degrade communication of the node under attack with its legitimate neighbors. Finally, even if the adversary successfully mounts this attack, all the false links are adjacent to one node (D) – which might be easily avoided by a node-disjoint route selection algorithm, e.g. [10].

5.4 Performance Evaluation in Benign Setting

To evaluate the performance of our protocol in a benign setting, we performed the following simulations. We distributed from 80 to 480 nodes uniformly in a field measuring 400 × 400m, assigning the nodes' transmission range to be 100m. We repeated this 1000 times. The unit disk graph (UDG) model has been used for determining neighbors of nodes. The values of the maximum distance estimation error e (as percentage of maximum range R) reflect, and even overestimate, the values we obtained in our experiments (Section 6.2). The value of ϵ_{quad} is set to $2e$.

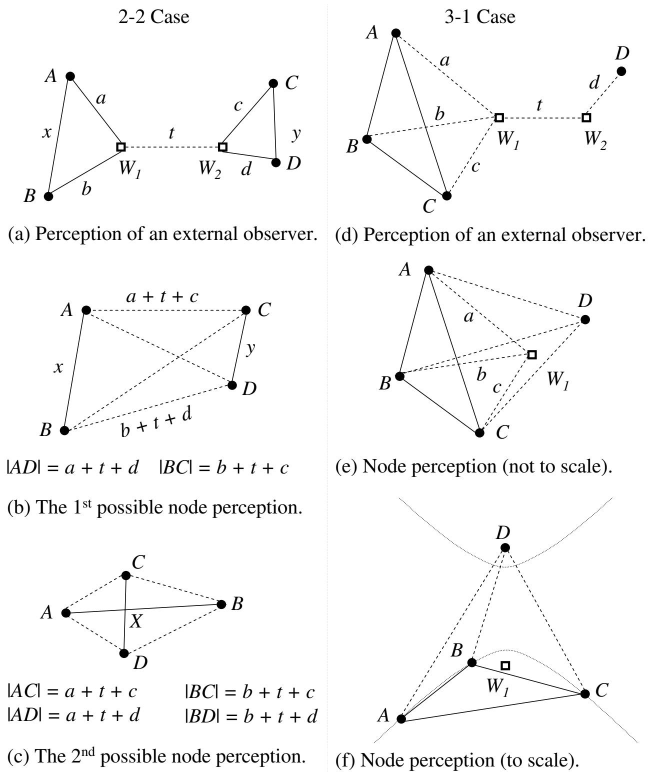
2-2 Case
3-1 Case
A
B
C
D
W1
W2
a
b
c
d
t
x
y
(a) Perception of an external observer.
(d) Perception of an external observer.
a + t + c
b + t + d
|AD| = a + t + d |BC| = b + t + c
(b) The 1st possible node perception.
(e) Node perception (not to scale).
X
|AC| = a + t + c |BC| = b + t + c
|AD| = a + t + d |BD| = b + t + d
(c) The 2nd possible node perception.
(f) Node perception (to scale).

Figure 2: 2-end Wormhole, 2-2 and 3-1 attacks. Circles denote correct nodes, squares denote wormhole ends, continuous lines denote real links, and dashed lines denote links created over the wormhole. Rather than the actual distance between W_1 and W_2, t represents the distance corresponding to the (non-negative) delay that the adversary chooses to introduce on link (W_1, W_2).

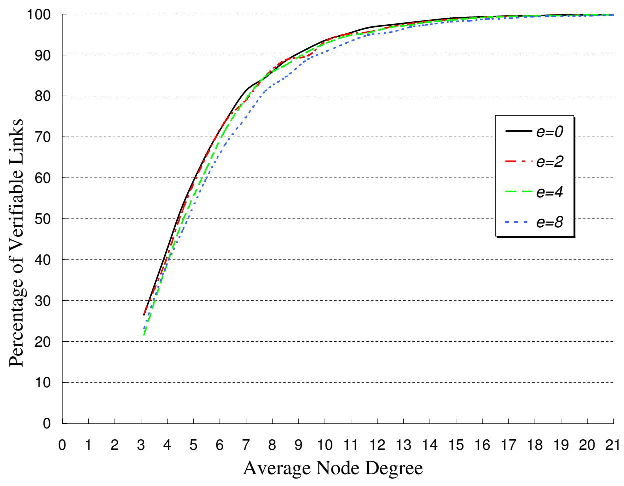

Figure 3: Coverage of the protocol. e denotes the maximum error in distance measurement (as percentage of maximum range R).

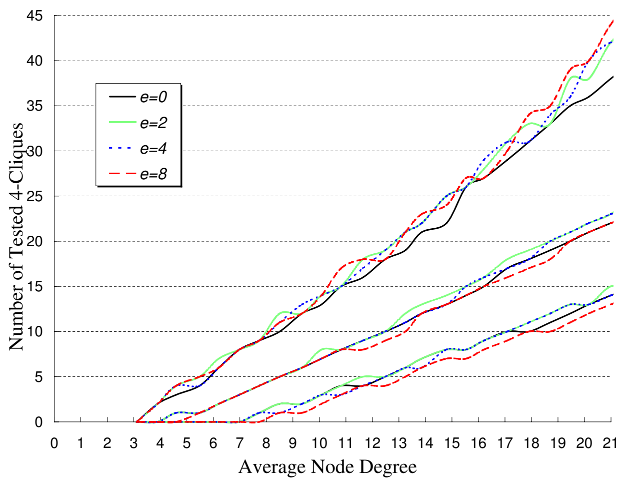

Figure 4: Computational complexity of the protocol. Three sets of lines, from top to bottom, represent 95%-percentile, median, and 5%-percentile of the number of the tested 4-cliques, respectively.

Coverage. Links have to satisfy the convex quadrilateral test to be verified by our protocol. Yet, even in a benign setting, some links might not belong to any convex quadrilateral, and therefore remain unverifiable. The *coverage* of the verification protocol is defined as the percentage of verifiable links. Figure 3 shows the coverage of the protocol vs. the average degree of nodes (i.e., average number of nodes that can be ranged by each node). For the networks with average node degree above 7, the coverage is more than 90%. We also note that only a negligible fraction of links cannot be verified due to the distance estimation error.

Communication Complexity A protocol run requires a node to transmit 3 RF messages (REQ, REP, ACK) to each of its neighbors, and to broadcast one US (RNG) and one RF message (NTE) to all neighbors. Hence, the protocol's communication complexity is constant per link.

Computational Complexity Testing whether a 4-clique is a convex quadrilateral, is by far the most computationally expensive component of our tests, explained in Section 4.3. As such, to estimate the computational complexity of the protocol, we count the "number of tested 4-cliques" per node. Figure 4 shows the number of tested 4-cliques vs. the average degree of nodes, for various distance measurement error values e. As it is shown, the number of tested 4-cliques for each node is almost linear to the number of its neighbors (i.e., constant per link).

6. IMPLEMENTATION

In this section, we describe the implementation of the ranging protocol on sensor motes and achieved experimental results. The security tests are not implemented on the sensor motes. The experimental results confirm the applicability of our protocol.

We used *Cricket* software v2 as the basis of our project and *TinyOS* v1 [6] as the operating system installed on the Cricket motes. To make the ultrasound signal omnidirectional, we simply mounted a metal cone on top of the motes to make the signals omnidirectional (similar to [19]).

6.1 Protocol Implementation

Neighbor Discovery. We implemented neighbor discovery as a simple 3-way handshake mechanism. The neighbor discovery process runs once when a mote is turned on. Because the key distribution protocol is orthogonal to our work, we did not implement any key exchange protocol. Instead, for simplicity, we assigned the cryptographic keys to motes before the nodes deployment.

Ranging. Ranging is the core component of the protocol. After a given node starts ranging, authenticated REQ messages are sent repeatedly to all of its neighbors every 100 milliseconds. This time interval is large enough to receive the corresponding REP message from the demanded neighbor. The starting time of transmission and reception of all REQ and REP messages are recorded using (32-bit) microsecond local timers. The accurate reception time of the messages is computed, after considering the delay introduced by bit alignment and the interrupt handling of radio frames (as described in [8]).

After finishing the synchronization phase, the initiator generates a $150\mu s$ pulse of 40 KHz ultra-sound using the US transducer, as the signal for distance measurement. Then, the ultra-sound nonce, N^s, will be transmitted using a simple on-off modulation. Every 65 milliseconds, a similar US pulse is sent if the corresponding bit of the nonce is 1. Otherwise, no signal is transmitted. The interval of sending successive bits is large enough (65 milliseconds) to avoid the inter-symbol interference.

All receivers, upon receiving the ultrasound signal store the arrival time to be used for distance measurement. To reconstruct the ultra-sound nonce, the receivers repeatedly listen to the channel. And in each interval if they receive the signal it is considered as 1. In the end, the assembled nonce is compared with the expected one (sent in REQ) and the records are discarded in the case of a discrepancy.

Finally, the initiator node sends the ACK message separately to all of the neighbors. Upon receiving the ACK message, nodes are able to calculate the distance and update their neighbor table. To compute the distance, based on the sound time-of-flight, we also consider the effect of the reflective cone we use above the ultrasound transducer; it always adds $10cm$ to the measured distance.

6.2 Experimental Results

Between 10 and 15 nodes were deployed on the floor in different configurations in a 4×4m room without obstacles in between. We placed nodes in different positions and, after running the protocol for several runs, results were processed off-line. To study to what extent the reflected cone effects the accuracy of the measurement, we ran the protocol also when nodes were located in positions where they could directly face each other around a circle. For each configurations we let the protocol run up to 70 rounds. Next, we present the results of the experiments that confirm the applicability of our protocol.

We investigate the applicability of the protocol in terms of the *time synchronization*, *distance measurement*, and *link symmetry* errors, due to the wireless medium irregularity.

Time Synchronization Error. Analyzing the results of all synchronization data (time-stamps) during the experiments, we derived that 99.55% of them met the constraints and only 0.45% of the links were discarded because the synchronization error was more than a $\epsilon_{sync} = 5\mu s$ threshold.

Distance Measurement Error. Nodes could reliably range each other for distances up to 4 meters when they were deployed on the floor with the cones on top of the transmitter unit. In these conditions, we achieved a maximum error of 5cm, where errors were positive values. In the face to face deployment, the maximum transmission range was 10 meters with maximum error of 3cm.

Link Symmetry Error. For each pair of nodes, A and B, we computed $|d_{AB} - d_{BA}|$ as the link symmetry error. This is used to adjust both ϵ_{sym} and ϵ_{quad} parameters. The observed symmetry error was below 7cm for 97% of the cases, and it was below 2cm for 74% of the cases. When the motes where placed in a circle of 4m diameter, facing each other, a maximum symmetry error of 2cm was experienced for 97% of the links.

7. CONCLUSION AND FUTURE WORK

We have designed a *secure neighbor verification* protocol tailored for *wireless sensor networks*. To demonstrate its applicability to WSN, we have provided a proof-of-concept implementation on existing off-the-shelf hardware (Cricket motes). We have proved that the protocol is secure against the classic 2-end wormhole attack. Yet, our scheme is also effective against more complex relay attacks, as we have demonstrated with simulations in the associated technical report [15].

Our scheme can be extended. For example, the scheme can be relatively easy augmentable to 3D networks by replacing the quadrilateral test with its equivalent in the new setting. Moreover, other distance estimation techniques are worthy of being implemented as (probably better) alternatives for US-based measurement.

8. REFERENCES

[1] H. Alzaid, S. Abanmi, S. Kanhere, and C. T. Chou. Detecting wormhole attacks in wireless sensor networks. Technical Report, Computer Science and Engineering School - UNSW, The Network Research Laboratory, 2006.

[2] L. Buttyán, L. Dóra, and I. Vajda. Statistical wormhole detection in sensor networks. In R. Molva, G. Tsudik, and D. Westhoff, editors, *ESAS*, volume 3813 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 2005.

[3] J. Eriksson, S. V. Krishnamurthya, and M. Faloutsos. Truelink: A practical countermeasure to the wormhole attack in wireless networks. In *ICNP'06*, 2006.

[4] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *Symposium on Network and Distributed Systems Security (NDSS)*, 2004.

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

[6] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. Tinyos: An operating system for sensor networks. In *Ambient Intelligence.* 2005.

[7] R. Maheshwari, J. Gao, and S. R. Das. Detecting wormhole attacks in wireless networks using connectivity information. In *IEEE Conference on Computer Communications INFOCOM*, 2007.

[8] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of ACM SenSys*, 2004.

[9] F. Nait-Abdesselam, B. Bensaou, and T. Taleb. Detecting and avoiding wormhole attacks in wireless ad hoc networks. *Communications Magazine, IEEE*, 46(4), April 2008.

[10] P. Papadimitratos and Z. Haas. Secure Data Communication in Mobile Ad Hoc Networks. *IEEE JSAC, Special Issue on Security in Wireless Ad Hoc Networks*, 24(2):343–356, 2006.

[11] R. Poovendran and L. Lazos. A Graph Theoretic Framework for Preventing the Wormhole Attack in Wireless Ad Hoc Networks. *ACM/Kluwer Wireless Networks*, 13(1), 2005.

[12] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux. Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility. In *ACM ASIACCS 2008.*

[13] K. B. Rasmussen and S. Čapkun. Implications of radio fingerprinting on the security of sensor networks. In *International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*, 2007.

[14] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *WiSe '03: Proceedings of the 2nd ACM workshop on Wireless security.* ACM, 2003.

[15] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P. Hubaux. A Low-Cost Secure Neighbor Verification Protocol for Wireless Sensor Networks. Technical report, EPFL LCA-REPORT-2008-020, 2008.

[16] A. Srinivasan and J. Wu. A Survey on Secure Localization in Wireless Sensor Networks. *Encyclopedia of Wireless and Mobile Communications*, 2008.

[17] S. Čapkun, L. Buttyán, and J.-P. Hubaux. Sector: secure tracking of node encounters in multi-hop wireless networks. In *ACM workshop on Security of ad hoc and sensor networks*, 2003.

[18] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In *ACM WiSe*, 2004.

[19] K. Whitehouse, F. Jiang, A. Woo, C. Karlof, and D. Culler. Sensor field localization: a deployment and empirical analysis. Technical Report UCB//CSD-04-1349, Univ. of California, Berkeley, 2004.

[20] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transaction on Sensor Networks*, 2(2):221–262, 2006.

[21] S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 62–72, New York, NY, USA, 2003. ACM.