LCA-REPORT-2008-020 Oct 2008

A Low-Cost Secure Neighbor Verification Protocol for Wireless Sensor Networks

Reza Shokri, Marcin Poturalski, Gael Ravot, Panos Papadimitratos,
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). However, the nature of wireless communications makes attacks against ND easy: an adversary can simply replay or relay (wormhole) packets across the network and mislead *disconnected* nodes they communicate directly. Such attacks can compromise the overlying protocols and applications. Proposals 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, often only against an attack mounted by at most two adversarial nodes (a 2-end wormhole). In this paper, we address these problems, designing a secure *neighbor verification* protocol for *low-cost* and *constrained* Wireless Sensor networks (WSN). Our scheme relies on nodes' distance estimation and simple geometric tests, and it is fully *distributed*. We prove our protocol secure against the classical 2-end wormhole attack, and we show it makes it hard to mount a k-end wormhole attack ($k > 2$): For non-negligible yet limited chance of success, the adversary must know the locations of all victim nodes and deploy a relay (wormhole) node close to *each* victim. We provided a proof-of-concept implementation with *off-the-shelf* WSN equipment, Cricket Motes. The protocol is the first applicable method that achieves a high level of security for WSN against sophisticated wormhole attacks.

1. INTRODUCTION

Neighbor discovery (ND) provides the essential functionality for wireless devices to be able to discover others that are within range or can communicate directly, with across the wireless medium. This is a fundamental building block for wireless networking and a multitude of applications, routing being the most essential in the context of Wireless Sensor Networks (WSN). Nonetheless, the nature of wireless communications makes it easy to abuse ND. Consider two sensor motes A and B, out of each other's 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 in order to be retransmitted to B. The result is that A and B are misled, through a false link, they are neighbors whereas they are not. Consequently, such an adversary can control the victims' communication, or more generally, can control multihop communication by shortening communication paths and manipulating other system's operations. For example, once the network needs to transmit a time-critical alarm, the adversary is able to cut-off the fictitious links and prevents the detection of the event by the network's authority.

The importance of "wormhole" or "relaying" attack on wireless networks has been identified in the literature, especially 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 discovery* aims to thwart this attack, essentially, to identify as neighbors only nodes that are indeed neighbors. This can be achieved as a two-stage approach: first, potential neighbors are discovered in a typical manner; then, a *neighbor verification* protocol verifies which nodes are indeed neighbors, to achieve secure ND. However, a neighbor verification protocol does *not* have to be *complete*, i.e. able to discover and verify *all* actual neighbors. Nonetheless, it can *detect* an adversary trying to create false links[1].

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., discovering other nodes it can directly communicate with [17]. 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 our survey in Sec. 2 reveals, many of them are not generally applicable to WSN, either because they utilize specialized hardware or because they require additional infrastructure. Besides, those which are applicable offer relatively limited security, often focusing on adversaries mounting 2-end wormhole attacks.

In this paper, we take on the challenge of designing a powerful and secure neighbor verification protocol that adheres to the limited hardware capabilities of WSN. At the same time, we realize that a 2-end wormhole is in fact a limited type of relay attack: an adversary might mount $k > 2$ nodes and therefore relays traffic among correct nodes (victims) through the attack network. Our objective is to be resilient even against such a more powerful adversary that mounts what we term a k-end wormhole attack. Overall, there has been, to the best of our knowledge, no concerted effort to design secure ND protocols that address these two problems and achieve acceptable level of security in WSN.

[1]It is the task of an intrusion response system to find the best possible *reaction* to the detected attack, but this is outside of the scope of this paper.

We develop a solution to address precisely these challenges. We design a secure neighbor verification protocol that is practical for low-cost WSN motes, as demonstrated by our *implementation*. Nodes estimate their distance to others in their vicinity, and exchange information about their estimates. Then, a series of simple geometric tests is run by each node over this local neighborhood view it obtained, in order to detect topology distortions created by wormhole attacks. The attack is detected if the distortion is beyond what is expected to happen as a result of channel error. Tested neighbors that do not reveal any distortion and also pass the verification test are verified to be physical neighbors and thus communication neighbors.

Our results show these tests are highly effective. First, our scheme is provably secure [2]against classical *2-end wormhole*, no matter how fast messages of correct nodes can be relayed, and independently of the adversary knowledge (e.g., correct nodes' location). Through simulation results, we demonstrate our scheme makes the network highly resilient against k-end relay attacks. Without centimeter-precision knowledge of correct nodes' locations, which is hard to obtain, the adversary has only a negligible chance to create false links and avoid being detected. Yet, even with such knowledge, the attacker is extremely limited and needs to deploy approximately one relay node per victim. Our distributed solution is independent of the technology nodes use to estimate distance to other nodes. Nonetheless, our proof-of-concept implementation relies on *off-the-shelf* WSN equipment: Cricket motes that rely on *ultra-sound* (US) ranging. We emphasize on the effectiveness of our scheme that utilizes a secure ranging, tailored to our security requirements, even though the simple US ranging is susceptible to relatively simple attack [19]. Overall, our contributions in this paper are: (i) the first secure ND scheme tailored for low-cost wireless nodes, notably in a WSN, (ii) the provable security of the protocol against a 2-end wormhole, (iii) investigation of a more general k-end relay attack, (iv) the effectiveness of our protocol against such stronger adversaries, and (v) the implementation and experimental evaluation of our secure ND in a WSN.

We study existing secure ND schemes in Sec. 2. System model is described in Sec. 3. Consequently, Sec. 4 presents our protocol and in Sec. 5 we analyze its security and performance. Next, in Sec. 6 we explain the experimental results. Some issues are discussed in Sec. 7 before Sec. 8 presents the future work and concludes the paper.

2. RELATED WORK

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

The scheme proposed in [7] develops a wormhole attack detection using directional antennas, which are not common for WSN. RF-fingerprinting, i.e., recognizing a wireless transceiver based on unique features of the signal it generates, has been considered in [18]. However, the signal analysis that this scheme performs is not feasible for a typical sensor mote radio receiver. Geographical packet leashes [8] rely on location information to estimate distance between the nodes, and thus prevent wormhole attacks. However, node knowledge of *secure* location is a requirement hard to fulfil in many settings. In particular, secure localization schemes for WSN require additional infrastructure such as a number of location-aware beacons [20]. This kind of additional infrastructure is also the basis of a wormhole prevention scheme proposed in [16].

Another way to estimate distance between nodes is to estimate message time-of-flight between two nodes: either for the ultra-sound medium [13], or for RF, e.g., [8] or [21]. The former is not secure in general: An adversary can decrease the distance by relaying a slow US signal over a light-speed RF or wired relay link. However, in our scheme we leverage on the presence of multiple correct nodes to detect these attacks. The latter is problematic, especially in the context of sensor networks, given the light-speed propagation of RF waves. To precisely estimate the distance, nanosecond precision is required, 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 [12] for generic wireless networks, in [5] for 802.11, or [1] for WSN. Our protocol utilizes this method as well. The available precision allows to thwarts store-and-forward wormhole attacks as well as to prevent a more sophisticated 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 approximate distance measurements [22] or only connectivity information [3] have been 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 abnormal values of some statistics of the connectivity graph, but this requires assumption about the expected values of these statistics. Further, no information about the location of the wormhole is provided if one is detected.

A distributed scheme that relies on only connectivity information detects wormhole attacks by checking for *forbidden structures* in the connectivity graph [10]. A forbidden structure is a graph that is very unlikely to be observed as a subgraph of a legitimate connectivity graph, but often appearing under a wormhole attack: The precise form of these structures depends on the connectivity pattern. The method proposed in [10] demonstrates good performance (specifically high detection rates) for dense networks when evaluated (with simulations) under theoretical connectivity patterns (unit-disk model (UDG), quasi-UDG model and the TOSSIM model). However, the scheme has not been investigated in a real deployment, with a high irregularity of antenna patterns [24]. Under such conditions, the authors of [10] propose to empirically estimate the forbidden structures in a part of the network free from wormhole attacks. This can be a substantial overhead, and can prove impossible in some applications. On the contrary, although our scheme also has some deployment-specific parameters which should be estimated empirically, these parameters are non-essential, i.e., are only used by secondary security mechanisms. These mechanisms can be turned off if estimation is

[2]Except an arguably infeasible case that the adversary has to mount its node on top of victims.

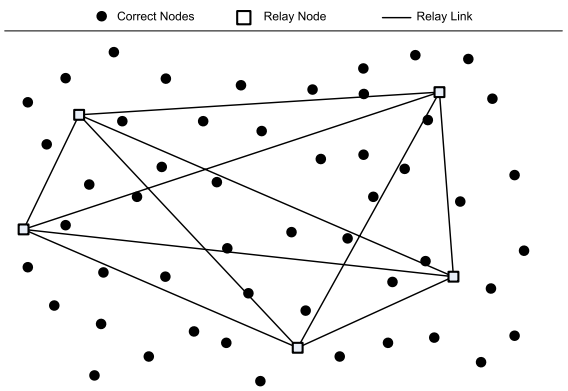


Figure 1: A 5-end Relay Attack.

not feasible. Furthermore, the security of this scheme is not analyzed under k-end relay attacks.

3. SYSTEM MODEL

We assume a static WSN with *sensor nodes* (*motes*), A, B, ..., distributed across 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 two network interfaces: a radio-frequency (RF) and a sound (typically, ultrasound (US)) interface. R is the range of the US technology, $s = 342$m/s is the speed of sound, and $c = 3 \times 10^8$ 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}. Symmetric keys can be established by one of the many key distribution schemes in the literature, as detailed in Sec. 4.

Adversary Model We are mainly concerned about *external* adversary that cannot compromise correct nodes or their cryptographic keys. The adversary controls a *relay network* composed of a number of *relay nodes* that we also term *wormhole ends*. The relay nodes are fully connected by out-of-band *relay links*, over which information propagates with speed equal to c. In our model, a relay node represents a pair of exactly one radio (RF) and one ultrasound (US) antenna. We model a sophisticated relay node that can *simultaneously* send and receive US signals (without causing interference on its own receiver) as two relay nodes. In other words, each attacker's US-RF antenna (directional or omnidirectional) is considered as a relay node.

If the regions under the control of the adversary, that is, the regions within range of the wormhole ends, do not overlap (i.e. no valid link exists between correct nodes in different regions), we term the attack a *remote* attack. Moreover, we assume the adversary to be capable of relaying messages on a *per-symbol* basis (i.e. messages are relayed in physical layer and the adversary is not limited to *store-and-forward*).

The processing time at the receiving and sending relay nodes is considered to be below 1μ sec. As the reception time of a US message is estimated with microsecond precision, such relaying delay has negligible effect on US-based distance measurement. This is so when the attacker relays US signals over short links. However, long relay links (above 300m) introduce non-negligible delay (a few microseconds) to 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.

We distinguish a number of parameters characterizing the adversary. First, the size of the relay network: a *k-end* attack is an attack that involves k relay nodes. An example of a relay network consist of 5 nodes is depicted in Fig. 1. The adversary can be *topology aware*, meaning that it knows the locations of all correct nodes (in the vicinity of wormhole ends); otherwise it is *topology oblivious*. Finally, a topology-aware adversary might mount a *post-deployment* attack that enables it to deploy the relay network knowing the physical position of correct nodes after they have been deployed in the field or it might be forced to deploy the relay nodes without this knowledge; i.e. mounting a *pre-deployment* attack, before the correct nodes have been placed at their positions.

4. THE SCHEME

After deployment, nodes establish *security associations* (SAs): a key establishment protocol, with numerous proposals in the literature, e.g., [6, 2, 4, 25], installs symmetric pairwise keys for communicating nodes. At this stage, there is no guarantee that nodes that appear neighbors are indeed so. Our secure neighbor verification protocol will address this, relying on the SAs. The protocol is intended to run after most of the WSN nodes are deployed and SAs established, even though false links naturally can exist. The higher the network density, the higher will be the effectiveness of the verification protocol. If the network topology changes significantly, by the operator relocating nodes or deploying new nodes, the protocol is re-executed. Basically, the protocol consists of a number of phases we describe in detail further below in this section:

(*i*) **Ranging** Given a list of potential neighbors, every node attempts to calculate its distance to each neighbor using ultrasound ranging. Ranging is done in a *secure* manner, meaning that adversarial behavior is restricted to the *relaying* of messages via the relay network.

(*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.

(*iii*) **Link Verification** With the information obtained during the neighbor table exchange, the node constructs a 2-hop neighbor table, which then locally (i.e. without further communication with other nodes) undergoes a battery of consistency tests. Consequently, an attack can be *detected*, and links can be *discarded* or *verified*.

The protocol operates with a set of five parameters ad-

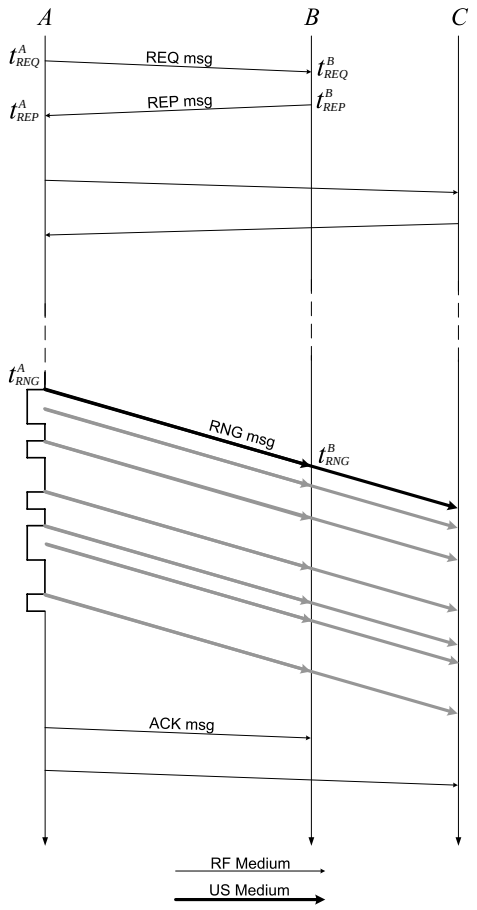


Figure 2: Ranging Protocol. The *initiator* node A sends the REQ message to each of its neighbors, here B and C. Each neighbor replies to complete the synchronization. Then, the RNG message is sent over the US medium. Finally, the information in the ACK message allows neighbors of A to verify clock synchronization and ranging and to calculate the distance based on the US time of flight.

justed based on the physical characteristics of the communication channel. θ_{sym} represents the acceptable fraction of links with asymmetric length (distance between incident nodes) due to the US signals irregularity. θ_{range} represents the acceptable fraction of links reported to be longer than R, due to the distance measurement error. ϵ_{sym} is a bound on the amount of link asymmetry error. ϵ_{quad} is the 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 error estimation. The values of these ϵ bounds are tradeoff between security and false alarm (and miss) rate. We choose rather conservative values, such that practically no false alarms happen. Moreover, all ϵ values are deployment-independent, and it will be shown later on we can harmlessly consider worst-case constant θ values. Aa a result, *our scheme is deployment independent.*

4.1 Ranging

A three-step *ranging protocol* is executed once by every node in the network. The first step of the ranging phase allows the *initiator* node, A, *synchronizing* with its (potential) neighbors (with microsecond precision) and is performed over RF. This is not really synchronization; rather, the difference between clocks readings at A and a potential neighbor B is determined by both A and B. The second step, performed over US, is the actual *ranging*. This operation is done *simultaneously* for all neighbors, significantly limiting the adversary behavior if it wants to remain undetected. The last *acknowledgement* step finalizes the ranging and secures the synchronization. Fig. 2 illustrates one execution of the ranging protocol between node A and two of its neighbors B and C.

Synchronization The initiator, A, first sends via RF a REQ message to neighbor B. The REQ message contains an encrypted freshly generated nonce: N^r_B, which is different for every neighbor B and is used to authenticate the response from B. It also contains the hash function value of a freshly generated nonce N^s, used for ranging, 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 t^B_{REP}. When A receives the REP message, it records the time of reception t^A_{REP}, to be used in the acknowledgement step. This procedure is repeated for each neighbor of A.

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

Ranging The initiator A, broadcasts the ranging message RNG over US and records the transmission time t^A_{RNG}. The message consists of a string of bits: a single digit 1 concatenated with the nonce N^s generated in the synchronization step. The purpose of transmitting a single digit 1 is to act as preamble and simplify the detection of the message start time at the receiver. Every neighbor B receiving this message records the time of reception t^B_{RNG}, along with the received nonce 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 every neighbor *that has replied correctly in the synchronization step.*

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

The ACK message contains N^s, to bind it to the earlier transmitted REQ and RNG messages, and three timestamps: t^A_{REQ}, t^A_{REP} and t^A_{RNG}. They allow B to check if 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. It also checks if the received nonce in the ranging phase, N_B^s, is equal to N^s (that was sent by A). If checks are successful, B uses the time-stamps to calculate the distance to A. To provide authentication and integrity, the ACK message also includes 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* and *distances* to all neighbors that were previously properly ranged. 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. Every node A combines its own NT_A and the received neighbor tables from its neighbors into the *2-hop neighbor table* $NT2_A$, to be used in the link verification phase.

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

4.3 Link Verification

The link verification consists of three steps as part of which a node A runs consistency tests on the 2-hop neighbor table $NT2_A$. If at any point a link is discarded, it is ignored in subsequent steps. The protocol will continue the verification even after an attack is detected, so that information on all discovered links is obtained and then utilized to analyze the attack (e.g., for intrusion detection). The link verification three tests are:

(*i*) **Link Symmetry Test**. Any link (U, V) for which the distance measured by U is different from the distance measured by V is discarded, which applies also to links for which only one measurement is available. If a fraction of more than θ_{sym} such links exists, an attack is detected.

for all $U, V \in NT2_A$ **do**
  if $|d_{UV} - d_{VU}| > \epsilon_{sym}$ **then**
    remove links (U, V) and (V, U) from $NT2_A$
    increase Cnt_{sym}
if $Cnt_{\text{sym}}/(\#\text{links in } NT2_A) > \theta_{\text{sym}}$ **then**
  attack detected

(*ii*) **Maximum Range Test**. Links which are longer than the range R are discarded. An attack is detected if a fraction of more than θ_{range} links is discarded.

for all $U, V \in NT2_A$ **do**
  if $d_{UV} > R$ **then**
    remove links (U, V) and (V, U) from $NT2_A$
    increase Cnt_{range}
if $Cnt_{\text{range}}/(\#\text{links in } NT2_A) > \theta_{\text{range}}$ **then**
  attack detected

(*iii*) **Quadrilateral Test**. In the last step, A looks at every 4-clique[3]in $NT2_A$ to which it belongs. If any of them is not a *quadrilateral* within ϵ_{quad} error tolerance (i.e. it is a tetrahedron with positive volume), an attack is detected. Moreover, a link is declared *verified* if it is a part of a *convex* quadrilateral.We note that after all the 4-cliques have been checked, some links might not be discarded, but not declared verified neither. We explain in Sec. 7 how these links can be treated. Next, we describe the components of quadrilateral test in more details.

[3]Note that as this is a clique and that A knows the lengths of all 6 links.

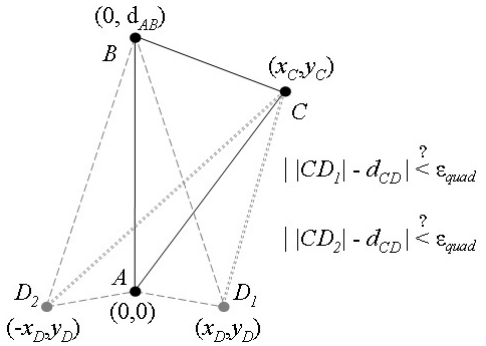


Figure 3: Quadrilateral Test.

for all $B, C, D \in NT2_A$ **do**
  if $\{A, B, C, D\}$ is a 4-clique **then**
    if $\{A, B, C, D\}$ fails *Quadrilateral Test* **then**
      adversary detected
    else if $\{A, B, C, D\}$ is *convex* **then**
      mark all links in $\{A, B, C, D\}$ *verified*

Quadrilateral Test. To test if a 4-clique forms a quadrilateral, one can check if it is possible to assign (locally) positions to the nodes such that no contradiction arises. This can be done as follows (Fig. 3): Assign position $(0, 0)$ to A, $(0, d_{AB})$ to B; check if d_{AB}, d_{BC}, d_{CA} form a triangle, i.e. if all 3 triangle inequalities hold. If not, there is a contradiction. Otherwise, assign position (x_C, y_C) to C, where $y_C = (d_{AB}^2 + d_{AC}^2 - d_{BC}^2)/(2d_{AB})$ and $x_C = (d_{AC}^2 - y_C^2)^{\frac{1}{2}}$. Further, 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 $y_D = (d_{AB}^2 + d_{AD}^2 - d_{BD}^2)/(2d_{AB})$ and $x_D = (d_{AD}^2 - y_D^2)^{\frac{1}{2}}$. For both positions, 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 a quadrilateral. Otherwise, assign to D the matching position. It is worth mentioning that, the ϵ_{quad} is calculated based on the propagation of distance measurement error in the this test.

Convexity and Co-Linearity Test. Given the positions computed in the quadrilateral test above, we test if the quadrilateral is convex whereas none of its 3 points are collinear. For the former, 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 it is positive, the quadrilateral is convex, if it is negative it is concave. To test the latter, check if the absolute value of any of these cross products is lower than ϵ_{quad}.

5. SCHEME ANALYSIS

In this section, we analyze the security of our scheme, in presence of 2-end and k-end attack, and also evaluate its performance in a benign setting.

5.1 Ranging

For this part of the analysis, we assume A and B are correct nodes executing the ranging protocol, with A being the initiator. The adversary is a topology-aware, post-deployment k-end relay network, and knows any key except

the key shared by A and B. Under these assumptions, 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 $d_{BA} \approx (|AW_A| + |W_BB| + \tau)$, where W_A and W_B are some relay nodes close to A and B, respectively, and τ is the distance corresponding to the additional delay that the adversary imposes when relaying over the relay network).

(r2) The protocol prevents the creation of very long false links, i.e., links that span more than $0.5c \times \epsilon_{sync}$.

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

According to the ranging protocol, if B calculates the distance, the following has happened (in this order, t_i are in global time):

(1) At time t_1, recorded as t^B_{REQ} by B, 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} by B, node B has sent message $m_2 = \langle REP, N^r_B\rangle$;

(3) At time t_3, recorded as t^B_{RNG} by B, 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}, node A has received the REP message to its REQ message sent at t_0. The only acceptable REP message is m_2, and because the nonce N^r_B is fresh and encrypted in m_1, only B can generate this REP message. Hence, $t_{2.1} > t_2$ and further $t_0 < t_1 < t_2 < t_{2.1}$. Combined with the fact that the synchronization test was successful in (4), this allows to conclude that within an acceptable bound ϵ_{sync} at time t_1 the clock of A is equal to t^A_{REQ} and the clock of B is equal to t^B_{REQ}.

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 $0.5c \times \epsilon_{sync}$ the ranging will fail, and link (A, B) will be discarded. Further, the store-and-forward adversary, which cannot replay a message before it entirely receives it, is detected 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 any relay node to send m_3 before some relay node receives m_3 sent by A at $t_{2.2}$. Moreover, the integrity of m_4 is protected, hence B learns the actual time of sending m_3 measured with A clock: t^A_{RNG}. As we have shown that the synchronization has been successful, 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 B calculates is either $|AB|$ (propagation over the real link), or $|AW_A| + |W_BB| + \tau$, where W_A and W_B are some relay nodes close to A and B, respectively, and τ is an additional delay that the adversary might choose to impose (propagation over the relay network).

5.2 Symmetry Test

We show that *for a given relay link* the adversary, to avoid being detected by the symmetry test needs to introduce a *single, constant* (non-negative) delay in both directions. Indeed, 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 Maximum Range Test

In this section, we quantify the effectiveness of the maximum range test. Consider a single relay link (W_1, W_2). Without the maximum range test, any pair of nodes A (neighbor of W_1), B (neighbor of W_2) in the range of the wormhole ends (possibly more than R if the adversary uses superior antennas) could be deceived into believing they are neighbors. The maximum range test limits the number of vulnerable pairs (A, B) by checking $|AW_1| + |BW_2| \leqslant R$. Therefore, the maximum range test prevents the creation of a false link if the distance from the respective relay nodes of either A or B is above R. Moreover, the test limits by 83.33% the fraction of vulnerable pairs among those with both correct nodes within R of the relay nodes.

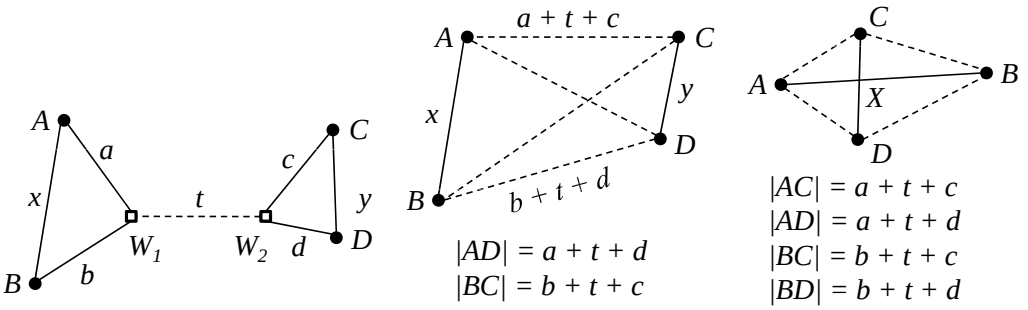
$$Pr((X + Y) \leqslant R) = \int_0^R \frac{2x}{R^2}\left(\int_0^{R-x} \frac{2y}{R^2} dy\right)dx = 16.67\%$$

where X and Y are the random variables representing distance of (nodes') positions to W_1 and W_2, respectively.

5.4 Security against 2-end Wormhole

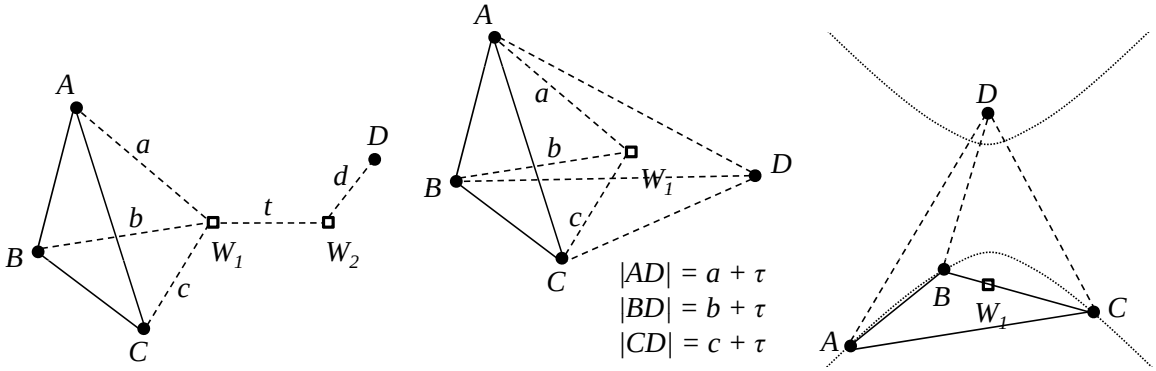
In this section, we investigate the security of our scheme against the most commonly considered relay attack: *single wormhole*, that is a relay attack utilizing two remote wormhole ends. Further, we limit the analysis to the case where nodes in communication range R are always able to communicate. Nevertheless, we consider the powerful external topology-aware adversary that mounts a post-deployment attack.

It is easy to show that an attempt to relay 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



(a) Perception of an external observer. (b) Node perception (case 1). (c) Node perception (case 2).

Figure 4: 2-end Wormhole, 2-2 Attack. Circles denote correct nodes, squares denote wormhole ends, continuous lines denote real links, dashed lines denote links created over a wormhole. Rather than the actual distance between W_1 and W_2, t represents the (non-negative) distance corresponding to the delay that the adversary chooses to introduce on link (W_1, W_2).



(a) Perception of an external observer. (b) Node perception (not to scale). (c) Node perception (to scale).

Figure 5: 2-end Wormhole, 3-1 Attack. Circles denote correct nodes, squares denote wormhole ends, continuous lines denote real links, dashed lines denote links created over a wormhole.

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 Fig. 4(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 (Fig. 4(b)). In this case:

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

However, in a quadrilateral (of positive volume) the sum of diagonals (|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 (Fig. 4(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 Fig. 4(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 Fig. 4(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 the adversary can create false links between at most one node from one side, but with an arbitrary number of nodes from the other side, if and only if this single node is located on top of the wormhole end (zero distance) and the other 3 do not circumscribe the other side of the wormhole.

Fig. 5(a) illustrates this attack. If $t + d = 0$, which means that the wormhole end is located in the same place as node

D, and that the adversary adds negligible delay when relaying the message, then 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.

Next, assume that $\tau = t + d > 0$. Assume further the adversary can convince the correct nodes that $ABCD$ is a quadrilateral. This means that AD, BD and CD intersect in a single point (Fig. 5(b)). 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 (Fig. 5(c)). As this hyperbola arm is concave, no 3 points A, B, C can form a convex quadrilateral with point D and hence validate false links AD, BD or CD. Hence no false links can be verified.

Considering the attack in the $t+d = 0$ case, it's applicability and effectiveness are limited. First, a negligible relaying delay is required. Second, deploying a relay node at a distance less than a few centimeters from a sensor mote might be difficult considering their physical appearance. Moreover, such 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 created links are adjacent to one node – which is easily circumvented by a node-disjoint route selection algorithm, e.g. [14].

5.5 Security against k-end Wormhole

In case of the k-end attack, although our protocol restricts the adversary, it is possible to create false links. In this section, we investigate via simulations to what extent our scheme can effectively detect the attack. As in the previous section, we consider nodes in range R to be communication neighbors. We analyze different types of adversaries: topology-aware and topology-oblivious.

Every experiment is repeated for a range of network densities, translating to average nodes' degree from 3 to 20. For each density, 1000 random *topologies* were tested. A topology is a random deployment of nodes in the vicinity of relay nodes. We assume that ϵ_{quad} is $0.04 \times R$ (a pessimistic value, much larger than the one we achieved in Sec. 6.2). For simplicity, parameters θ_{sym} and θ_{range} were set to 1; therefore, effectively, detection is limited only to the quadrilateral test, although all the test have been used for attack prevention.

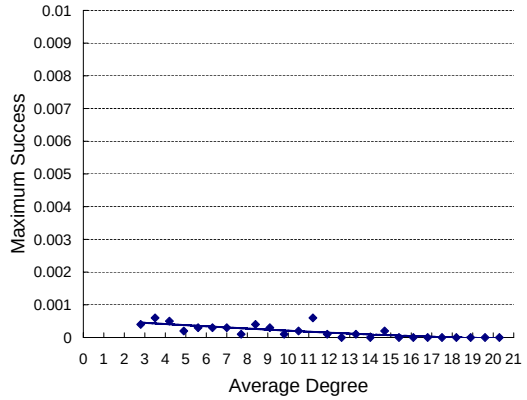
Simulations were performed for 3-end and 4-end attacks. Relay nodes can introduce positive relaying delay when relaying the RNG messages. We assumed the adversary is sophisticated enough to apply the same delay on relayed messages in both directions of each relay link, thus avoiding detection with the link symmetry test. Hence, an adversarial *strategy* is a sequence of 3 (3-end) or 6 (4-end) constant delays selected for each relay link. A strategy is *successful* if the adversary can create at least one verified false link without being detected by the protocol. As the space of all strategies is too large to perform an exhaustive search, we used *random selection* sampling method. Thus, for each topology, 10^4 strategies were investigated.

Fig. 6 illustrates the effectiveness of our protocol in the presence of 3-end and 4-end *topology-oblivious* adversary. Least square algorithm is used to fit polynomial degree two on the results. The *detection rate* of the attack is an average (over all examined topologies) of the fraction of strategies that are detected by the protocol. To evaluate the effectiveness of our attack detection, topologies for which the adversary cannot make *any* links over the wormhole (due to maximum range test, described in Sec. 5.3) are not counted. The *success probability* of the adversary is computed as the average fraction of successful strategies, over all topologies. However, because the success probability of the adversary is extremely small (in the order of 10^{-6}), we show the maximum success chance of the adversary strategies among investigated topologies. The 4-end adversary has full control over all the false links, hence a higher chance to create false links over the wormhole. Nonetheless, it is more probable to be detected by the quadrilateral test, because the probability of violation from creating quadrilaterals also increases. The 3-end adversary is more limited because it is restricted to 3 parameters in its strategies and also has control over 5 links in a 4-clique, hence a lower chance to even construct a quadrilateral. The simulation results confirm this, as the 4-end attack is slightly more successful than the 3-end attack. Finally, for an adversary with more than 4 relay nodes, the chance of success would decrease as the number of relay node increases. Without the knowledge of precise nodes' location, the adversary is very likely to be detected.

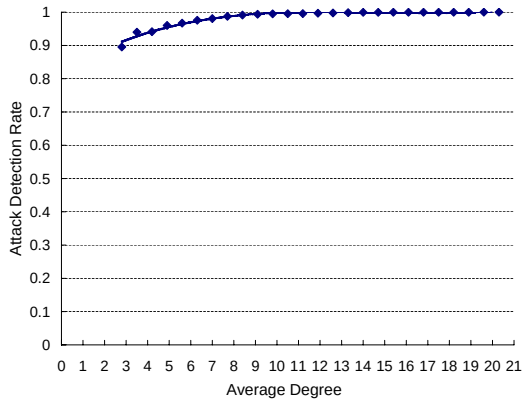
According to the results, it is very probable for the topology-oblivious adversary to be detected. Moreover, the success probability of the attacker is extremely low. To evaluate more powerful adversaries, in the following, we estimate the effectiveness of our protocol in presence of 4-end topology-aware adversary. The adversary searches among the sampled strategies if there is any successful strategy for each topology. Then, the best strategy that maximizes the number of false links will be selected for perpetrating the attack. The success probability of the adversary is computed as the fraction of topologies for which there is at least one successful strategy. The detection rate of the protocol calculated as the number of topologies the attacker gets detected, over the number of topologies that adversary can make any false link. Because we do not examine all strategies, the results provided by these simulations are only estimations for the real rates. Figures 7(a) and 7(b) illustrate the result. As it is shown, the success probability of the attacker decreases as the density increases. This is because with increasing the average degree the chance of creating an incorrect quadrilateral, and hence detection, increases, and results in low success probability. Relatively low success in low density networks is due to the fact that in most of the topologies the adversary can not create any quadrilateral.

Still, the attack can be more sophisticated. Instead of relaying messages with default transmission power, a more sophisticated adversary can limit its transmission range (while relaying the RNG message) sufficient to cover only the closest nodes to its relay nodes, thus decreasing the chance of being detected and hopeful to create verified false links. To see what is the effectiveness of our protocol in presence of such an adversary, we simulated such a power control policy on a 4-end wormhole. The transmission range of each relay point considered to be the distance to the closest node plus 10% of R (To make sure the victim receives the signals correctly). As it is shown in Figures 7(c) and 7(d), the attack is more successful using the power control technique. Nevertheless, the success of the adversary is limited due to the fact that the smaller area is covered.

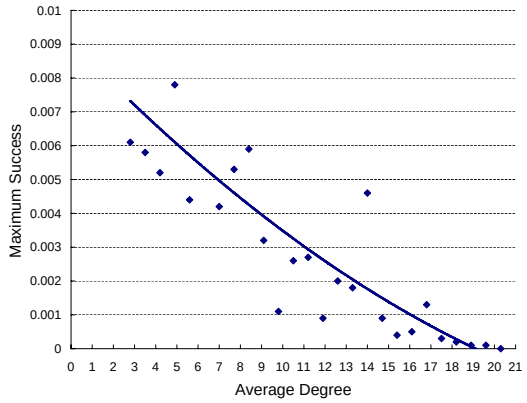
Thus far, we have not considered the attacker that mounts a post-deployment attack, knowing the location of correct nodes. Such a powerful topology-aware adversary that also



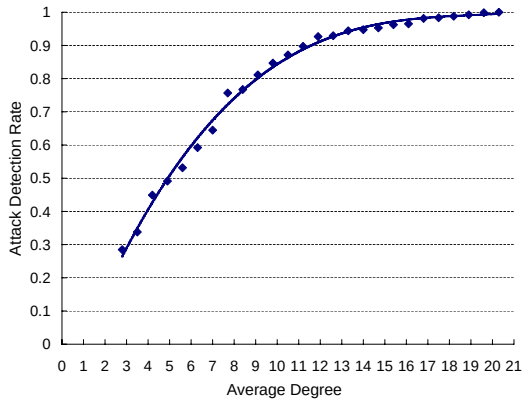
(a) Success Chance for 3-end Attack.



(b) Detection Rate of 3-end Attack.

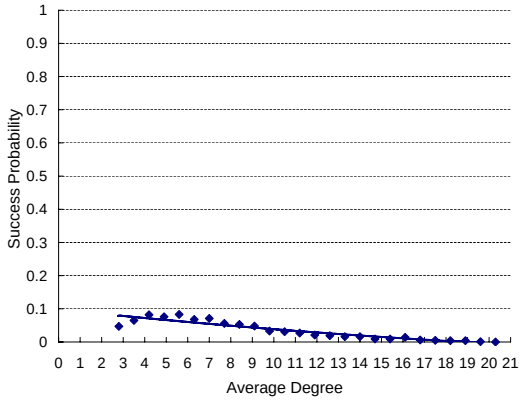


(c) Success Chance for 4-end Attack.

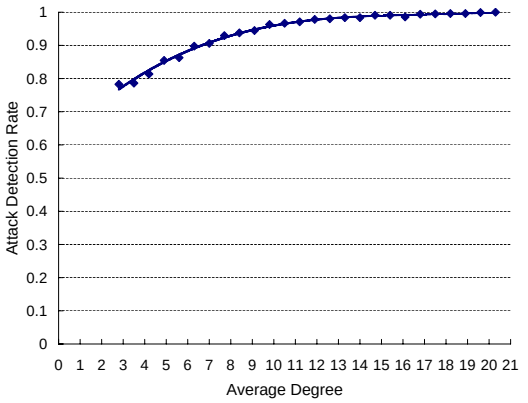


(d) Detection Rate of 4-end Attack.

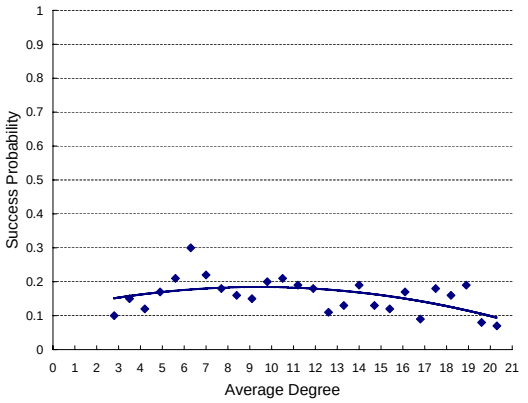
Figure 6: Topology-Oblivious Adversary



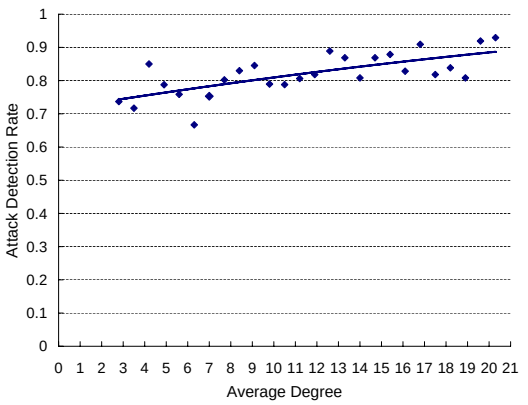
(a) Success Chance for Adversary.



(b) Detection Rate of Adversary.



(c) Success Chance for Adversary, with Power Control.



(d) Detection Rate of Adversary, with Power Control.

Figure 7: Topology Aware 4-end Attacker

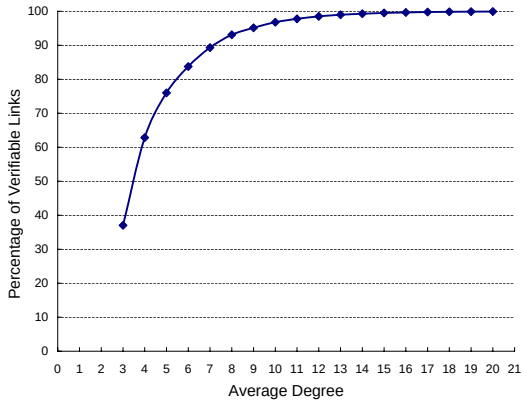


Figure 8: Coverage of the protocol.

controls its transmission power can create limited number of verified false links in the network, by placing a relay node close to 4 selected nodes, adjusting its power such that only these nodes will receive the relayed messages and then picking the relay links delay appropriately. Further in Sec. 7 we will discuss how hard implementing such an attack is and how we can disallow the adversary of using this capability with the help of other security protocols.

5.6 Performance Evaluation in benign setting

To be verified by our protocol, links have to satisfy the convex quadrilateral test. Yet, even in a benign setting, some links might not be an edge of any convex quadrilateral. Such links will remain unverified. We discuss in Sec. 7 how such links can be utilized, but in this section we evaluate the number of such links as a function of network density using simulation.

We distributed from 80 to 480 nodes uniformly in a field measuring 400×400. The transmission range of nodes is 100. We used the unit disk graph (UDG) model for determining neighbors of nodes.

The *coverage* of the verification protocol is estimated as the percentage of verifiable links. Fig. 8 shows the coverage of the protocol vs. the average degree of nodes. As it is shown, for the networks with average nodes' degree above 7, the coverage is more than 90% and the network is connected.

6. IMPLEMENTATION

In this section, we describe how we implemented the protocol on sensor motes. Then, we will show the applicability of our protocol through examining it in real deployment.

We used *Cricket* software v2 as the basis of our project and *TinyOS* v1 [9] as the operating system installed on the Cricket motes (Fig. 9). To make the ultrasound signal omnidirectional, we simply mounted a metal cone on top of the motes (Fig. 9) to make the signals omnidirectional (similar to what proposed in [23]).

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 a complicated key exchange protocol. Therefore, for simplicity, we assigned the cryptographic keys to motes before the deployment.

Ranging. Ranging is the core component of the protocol. After a given node starts ranging, authenticated REQ messages are sent repeatedly to all its neighbors every 100 milliseconds. This time interval is large enough to receive the corresponding REP message from the demanded neighbor. The sending and reception time of the first bit of data segment for 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 packets in radio board (as described in [11]).

After finishing the synchronization phase, the initiator generates a $150\mu s$ pulse of 40 KHz ultrasound using the US transducer, as the signal for distance measurement. Then, the ultrasound 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 will be transmitted. The interval of sending successive bits is large enough (65 milliseconds) to avoid inter-symbol interference.

All receivers, upon receiving the first part of the ultrasound signal will store the arrival time to be used for distance measurement. To reconstruct the ultrasound 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 discrepancy.

Finally, the initiator 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 considered the effect of the reflective cone we used above the ultrasound transducer, which always added $10cm$ to the measured distance.

Neighbor Table Exchange and Link Verification. After finishing the ranging, nodes are supposed to broadcast their Neighbor Table. In the current version of the implemented protocol, nodes simply send the neighbor tables to the base-station and security checks will be performed based on those results in a computer connected to the base station (just for simplifying the analysis).

6.2 Experimental Results

Between 10 and 15 nodes were deployed on the floor in different topologies in a $4 \times 4m$ room without any obstacle 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 extends the reflected cone effects the accuracy of the measurement, we ran the protocol also when nodes were located in positions that (their US transceivers) could directly face each other around a circle.

In all our experiments, ranging command were sent frequently from BS to the network. For each experiment we let the protocol run between 40 and 70 rounds. Here, 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

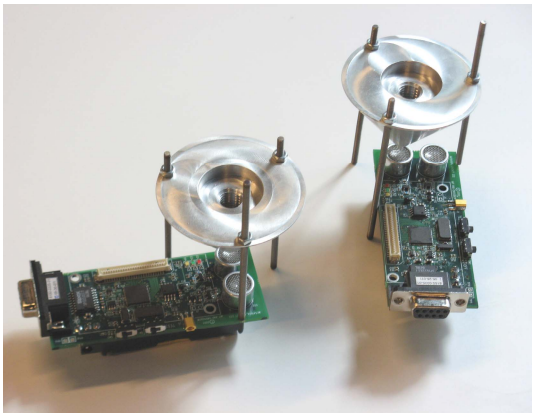


Figure 9: Crickets with reflective cones above the ultrasound transducers.

of all synchronization data (time-stamps) during the experiments, we learned that 99.55% of them met the constraints and only 0.45% of cases discarded because of synchronization error was more than a $\epsilon_{sync} = 5\mu s$ [4] 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 $5cm$. In face to face deployment, the maximum transmission range was 10 meters with error below $3cm$.

Link Symmetry Error. For each pair of nodes, we compared the difference between the distance estimated by two end nodes for their link as the link symmetry error. This is used to adjust both ϵ_{sym} and ϵ_{quad} parameters. The symmetry error was below $7cm$ for 97% of the cases (while it was below $2cm$ for 74% of the cases). When the motes where placed in a circle of $4m$ diameter, facing each other, the maximum symmetry error of $2cm$ was experienced for 97% of the links.

7. DISCUSSION

Unverified Links The protocol definition does not specify how to treat links that are neither discarded nor verified. The simplest and arguably most secure option is to discard them. However, these links could be beneficial to routing protocols, by increasing the diversity of the nodes topology view. Links in the neighbor table with an *unverified* status could have low but not zero probability of being selected as part of a route. This could be useful in low density networks, where the fraction of unverified links is significant, in contrast to strict rejection policy that might dampen or even prevent network-wide communication.

Topology-aware Adversaries Among all types of relay adversaries, topology-aware ones are the most powerful ones: they can deploy their relay network knowing the locations of correct nodes, and use power control to lower the risk of detection risk. However, knowing the network topology and node locations is not trivial. Even if the adversary manages to do this, its impact can be mitigated by an appropriate design. To reduce a topology-aware adversary into a topology-oblivious adversary, the system should force the adversary to deploy its relay nodes before the network deployment, thus disabling its choosing positions relative to those of correct nodes. The unpredictability of RF signals makes it hard for the adversary to infer precise locations before correct nodes commence ranging, thus, overall, the attacker would be topology-oblivious, unless of course the adversary obtained the network map by other means, e.g. manually. If protocols such as LEAP [25], which limit node shared key establishment within seconds from their being powered up at deployment, are used, the adversary must deploy its relay network in advance. This way, the adversary's power can be reduced to power of topology-oblivious one, with the drastically reduced effectiveness (Sec. 5).

Internal Adversaries In Sec. 5, we analyze the security of our scheme mostly against an external adversary, the type of adversary typically assumed in the literature for two reasons. A relay attack is naturally an external one. Moreover, internal adversaries cannot be, in general, defeated. Consider two corrupt nodes, A and B, with A sharing its credentials with B, allowing B to convince all its neighbors that A is a neighbor. In the light of this attack, it is reasonable to focus on links only between *two non-corrupted* nodes. However, many schemes, including ours, rely on information from additional nodes to verify links. Hence, even if the two nodes incident to a link in question are correct, false information from another node could potentially mislead them into falsely believing they are neighbors.

Due to the fact that sensor motes are inexpensive devices that can be tampered with, node corruption is feasible. For simplicity, we consider a relay network in possession of private keys obtained by corrupting m legitimate sensor motes. As shown in Sec. 5.1, the adversary cannot decrease the distance between two correct nodes. However, it can decrease the distance between a non-corrupted node and a relay node posing as a regular sensor mote (by sending the RNG message before t^A_{RNG} as declared in the ACK message). In addition, the adversary can also cheat during the neighbor table exchange phase, reporting arbitrary distances between two corrupted nodes. The creation of a few false links becomes easier: for example, a 2-end wormhole with 2 corrupted credentials can be used to convince two nodes they are neighbors. Nevertheless, a preliminary analysis hints that creating a large number of links, compared to the numbers of wormhole ends, k, and corrupted nodes, m, is not feasible, similarly to the external adversary case.

8. CONCLUSION AND FUTURE WORK

We designed a *secure neighbor verification* protocol tailored for *wireless sensor networks*. To demonstrate its applicability to WSN, we provided a proof-of-concept implementation on existing off-the-shelf hardware (Cricket motes). We proved that the protocol is secure against the classical 2-end wormhole attack. Moreover, through the simulations we demonstrated the infeasibility of mounting a successful k-end attack in the network. In fact, even such a powerful adversary is highly limited and is not able to make more than a few negligible fake links over the wormhole, comparing to the size of the attack network itself. Besides, the attacker will be detected with high probability, even though some fake links are made.

In future work, we intend to have a more extensive analysis of the protocol, notably for the topology-aware adversary, in order to estimate theoretical bounds on the attack's power. We also see potential for extending our scheme. For

[4] It corresponds to less than $2mm$ of sound propagation.

one, the scheme should be relatively easy augmentable to 3D networks by replacing the quadrilateral test with its equivalent one in new setting. Moreover, other techniques to estimate the distance between motes are worthy to be implemented as (probably better) alternatives for US-based measurement. Finally, developing a scheme that enforces higher level of limits on the internal adversaries would surely be worthwhile, but challenging as well; in fact, virtually all of the existing secure neighbor discovery proposals assume an external adversary, which hints about the difficulty of this problem.

9. 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] R. Anderson, H. Chan, and A. Perrig. Key infection: Smart trust for smart dust. In *Proc. of the 12th IEEE ICNP*, 2004.

[3] 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.

[4] W. Du, J. Deng, Y. S. Han, P. Varshney, J. Katz, and A. Khalili. A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks. *ACM Transactions on Information and System Security*, 48(2):228–258, 2005.

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

[6] L. Eschenauer and V. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In *Proc. of the 9th ACM CCS*, 2002.

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

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

[9] 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*, pages 115–148. 2005.

[10] 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.

[11] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49, New York, NY, USA, 2004. ACM.

[12] 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.

[13] U. S. Naveen Sastry and D. Wagner. Secure verification of location claims. Number UCB/CSD-03-1245, 2003.

[14] 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.

[15] 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. *IEEE Communications Magazine*, 46(2):132–139, February 2008.

[16] R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. *Wirel. Netw.*, 13(1):27–59, 2007.

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

[18] 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.

[19] S. Sedighpour, S. Čapkun, S. Ganeriwal, and M. Srivastava. Distance enlargement and reduction attacks on ultrasound ranging. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 312–312, New York, NY, USA, 2005. ACM.

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

[21] 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*, pages 21–32, New York, NY, USA, 2003. ACM Press.

[22] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 51–60, New York, NY, USA, 2004. ACM Press.

[23] 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.

[24] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):221–262, 2006.

[25] 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.