31

Neighbor Discovery: Security Challenges in Wireless Ad hoc and Sensor Networks

Mohammad Sayad Haghighi and Kamal Mohamedpour
K.N. Toosi University of Technology
Iran

1. Introduction

Wireless ad hoc and sensor networks are infrastructureless systems with self-configuration capabilities. Neighbor discovery protocols are fundamental requirements in the construction of self-organizing networks. Each computer (node) must discover who its neighbors are in order to be able to coordinate with them for any later communication. This goal is usually accomplished through broadcasting methods in the initial phases of network deployment.

Along with the development of neighbor discovery protocols, security threats also introduced and some researchers discovered new forms of attacks for neighbor discovery scenarios. Authors in this field have given different definitions on what a neighbor is and to what extent an adversary is equipped. We will first clarify these differences by making some categorizations and definitions before proceeding to the introduction of attacks and solutions.

After the introduction of initial concepts, we classify the attacks into two general groups and explain the solutions for each of the groups numerating the pros and cons of them. We will review the current external attacks solutions for the neighbor verification problem first and start with the early simple methods which tried to defeat the relaying attacks (like the wormhole one) using distance estimation methods. This family of protocols relied on time stamps which needed tight clock synchronization among the nodes and was quite impractical in distributed networks specially the sensor ones. After that, we introduce the descendants of that family of protocols which resolved the clock synchronization problem by using challenge-response-like methods.

Then, the recent efforts on formal description of the time-based and time- and location-based neighbor discovery protocols are explained. These researches led to the conclusion that time-based protocols can only secure the neighbor discovery under some strict conditions. Time- and location-based protocols are generally more secure than the time-based ones alone.

Next, we will argue why all of these protocols are vulnerable to the internal attacks and introduce other methods for defeating internal adversaries. Describing the mostly cryptographic solutions in this domain, we outline their pros and cons. As we will see, almost all of these protocols are either unable to resist the invasion of an internal adversary equipped with both powerful transmitter and sensitive receiver, or need an initial setup

phase in which the network is assumed to be secure. Adding the lack of mobility support to the specifications of this family of solutions, we move on to present our solution for a special type of internal attack which statistically tries to block the sensitive broadcasting internal adversary in mobile dense networks. This approach works based on the inherent characteristics of the dense networks and only the medium access control protocol parameters are changed. Therefore, it imposes a very low cost to the system to create higher levels of robustness.

At the end a conclusion is made which suggests combinational methods to be employed in the neighbor verification protocols to achieve an acceptable level of security against both the internal and external attackers.

1.1 Fundamental Concepts and Definitions

In wireless ad hoc networks, computers (nodes) are usually stand-alone entities working in cooperation with each other in order to fulfill a desired task (Rubinstein et al., 2006). In such networks, nodes are located centimeters to hundred meters away from each other but can communicate through their wireless transceivers.

Sensor networks can be thought to be a subgroup of ad hoc networks with some specific characteristics. They are usually deployed (densely) in an area to sense or monitor quantities of desired form (Akyildiz et al., 2002). For example in a battlefield, enemy movements can be detected and localized by a distributed sensor network. Unlike the ad hoc networks, the messages created by sensor nodes are always destined to be received by a single target named "Sink". The sensors cooperate to deliver the messages to the sink in a multi-hop manner.

The unattended nature of sensor networks and the high number of nodes creates some constraints for the designer. Each node must survive days or even years with a single source of power. Therefore, the protocols must not be too power consuming and should minimize both the amount of processing and the number of transmissions. The processing power and the amount of RAM[1] and ROM[2] of a sensor node are also quite limited. So, generally, designing a secure protocol is much harder in the sensor networks than in the ad hoc ones.

It is obvious that before a distributed cooperative network begins to work, nodes need to know their neighbors to form local structures. Any long-distance communication must be made in a multi-hop manner. The transmission radii of the nodes are limited and hence, each node should pass the messages to one/some of its neighbors in order to participate in the delivery of messages.

Neighbor discovery means determining whether a wireless device (node) is directly reachable without the assistance of any other device according to the predesigned rules of the network or not. Neighborhood can be unidirectional or bidirectional depending on whether only one side is able to deliver its messages to the other side or both sides are capable of doing so.

A neighbor-discovery attacker tries to deceivingly convince the nodes to believe that they are neighbors of a specific set of nodes (possibly including the adversary herself), when they are actually not. There are various types of attacks on neighbor discovery scenarios. The effectiveness level of an attack depends on whether the adversary is a part of the network or

[1]Random Access Memory
[2]Read-Only Memory

not. In the next sub-section, we make some definitions on adversaries' types and their level of capability.

1.2 Types and Capabilities of Adversaries

To describe the current solutions for the neighbor verification problem in the rest of this chapter better, we categorize them into two groups based on their resistance to internal (intrusive) or external (non-intrusive) attacks.

In external attacks the adversary is not able to compromise the nodes and hence, does not have access to the private information like the cryptographic keys and communication codes stored in the memory of the nodes whereas in the internal ones has (Khelladi et al., 2005).

Usually, an external attacker is only able to overhear (eavesdrop), relay (replay) or block (jam) the packets. On the other side, an internal attacker is capable of masquerading himself as a legal node and thus can imitate all the behaviors of a healthy node. Having the private cryptographic keys, she can even generate fake (but authenticated) messages to obtain a higher number of neighbors to what a traditional healthy node does. It is rather obvious that the second type of adversary is much more powerful than the first one.

1.3 Effective Attacks on Neighbor Discovery

In this part, we briefly introduce the currently known neighbor-discovery-related attacks in ad hoc and sensor networks. There are a few general attacks which have effects on neighbor discovery, and a few others which specifically address the neighbor-discovery-related issues.

One of the oldest external passive attacks is eavesdropping. Regardless of the protocol architecture, an adversary is always able to overhear wireless communications. There is little chance for the designer to block eavesdropping. However keyed cryptographic operators (like the encryption ones) are quite useful in keeping the external adversaries from extracting sensitive information out of the transmitted signal (Zhu et al., 2006)(Du et al., 2005)(Du et al., 2006). Neighbor discovery protocols are no exception. The protocol designer must seal the places where the information might leak during wireless transmissions.

The active invasions that target the availability of network services are called Denial of Service (DoS) attacks. DoS attacks can be planned to work on any layer of the network protocol stack depending on how much weak that layer is. Jamming can be well categorized into the physical layer DoS attacks group. There are only a few non-perfect classic solutions like spread spectrum communication for this attack (Pickholtz et al., 1982). Other types of DoS attack also exist among which some try to excessively overload a badly designed protocol run on a resource-limited machine (Djenouri et al., 2005). So, one can easily conclude that in sensor networks, designing a DoS-resilient neighbor discovery protocol is more complicated than in ad hoc networks. Ignoring the heuristic solutions, the classic countermeasures for protocol-related DoS attacks are easy-to-compute checksums and ciphers that reject massive fake messages.

Relaying and replaying are two other simple but powerful attacks (Papadimitratos et al., 2008). In the replay attack, an adversary uses an old packet which was previously generated by a healthy node in order to deceive another healthy node in the future. To overcome this problem researchers have suggested using timestamps (in clock-synchronized networks) and nonces (Du et al., 2006)(Shokri et al., 2008).

Relaying attacks are harder to detect. The adversary relays the healthy nodes' packets instantaneously (either at the physical layer or in a store-and-forward manner) in another part of the network. Wormhole attack is a well-known representative of this family. In the wormhole (tunneling) attack, two (or more) adversarial nodes try to transfer information through a dedicated channel between themselves and then use it in another part of the network (Hu et al., 2006). This attack can be implemented both by internal and external adversarial nodes. In the external form, they simply relay the packets either in a store-and-forward manner or instantaneously at the physical layer. In the internal form, they also have the opportunity to alter the packet contents intelligently before forwarding.
Every neighbor discovery protocol is composed of a series of packet transmissions. In a weak protocol, these attacks can be launched to relay neighbor discovery packets to other areas of the network, in order to convince distant nodes to believe that they are true neighbors. Figure 1 shows two healthy nodes *A* and *B*, and two adversarial nodes *C* and *D*. *A* and *B* cannot see each other directly since their transmission radii (shown with circles around them) are small compared to their distance. However, *C* and *D* can relay the neighbor-discovery-related packets to create a virtual link.

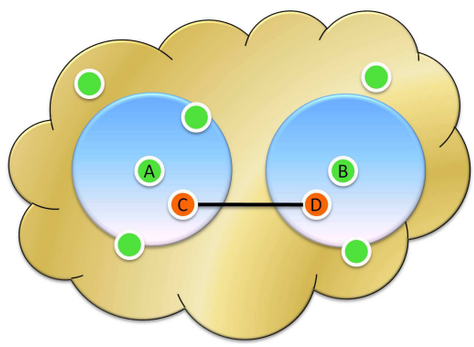


Fig. 1. The wormhole attack: Two adversarial nodes *C* and *D*, relay the packets between healthy nodes *A* and *B*, to deceivingly convince them to believe that they are in the vicinity of each other

Hello flooding is a broadcast-type attack that was originally designed for sensor networks (Karlof et al., 2003). However, its concept can be adopted in the ad hoc networks too. Most of the neighbor discovery protocols (as well as the routing protocols) use a broadcasted packet called "Hello" or "Beacon" to announce a node's presence to its neighbors. Nodes receiving this message assume that the sending node is one of their neighbors. An internal attacker can launch a hello flooding attack by simply broadcasting the hello message with very high power. This way she tries to convince a lot of nodes that she is one of their neighbors and the victims add her to their table of neighbors. If the adversary is well equipped, she might even have a low noise sensitive receiver which enables her to receive distant weak signals and thus turn this attack into a bidirectional one. Figure 2. shows the adversarial and healthy transmission ranges.
This attack was initially designed for the internal attackers. However a simple repeater-like external adversary can also launch a similar attack through boosting a healthy node's transmission power and acting as a "man in the middle". Fortunately, as we will see, there are more solutions for the external threats than the internal ones.

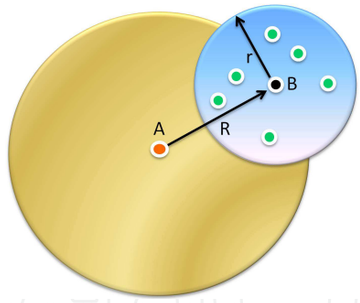


Fig. 2. A broadcast-type attack (hello flooding) diagram in a network with regular transmission range of r. The adversary (A) transmits messages with high power pretending to be a neighbor of B whereas she is actually R meters away ($R>r$)

Broadcast attacks can also be implemented in routing protocols in the route discovery phase (e.g. Routing Request (RREQ) message broadcasting) to shorten the adversary's path to the destination and thus putting her into most of the routes.

2. Neighbor Discovery External Attacks Countermeasures

If the maximum transmission range of a healthy node is r, then a secure neighbor discovery protocol must reject any claiming neighbor farther than this distance. The most challenging threat in designing externally resistant neighbor discovery protocols is the wormhole (or relaying) attack and the easiest way to block this attack is estimating the distance between the nodes.

The very early methods of neighbor verification relied on the round trip propagation time (RTT) measurement to estimate the approximate distance between two nodes and compare it with a maximum allowable value. Brands et al. were pioneers in using this method with their one-bit exchange proposal (Brands et al., 1993). The distance bounding method they proposed was a cryptographic protocol that put an upper-bound on the distance between two users (nodes for example) and did not let the protocol be manipulated. This scheme was able to resist man-in-the-middle like attacks, however, Singelee et al. later claimed that Brands' scheme is unable to stop what they called "terrorist fraud attack" which is an internal attack in our terminology (Singelee et al., 2005). After the introduction of distance bounding concept, similar protocols with names like "Echo protocol" (Sastry et al., 2003) appeared which were all based on the same basic idea of round trip time measurement.

In (Hu et al., 2003), to prevent wormhole attacks in ad hoc networks, a method called "packet leashes" was introduced and an idea similar to the signal trip delay measurement was repeated. The authors proposed two types of methods which could resist the (external) wormhole attacks: geographical leashes and temporal leashes. Geographical leashes fall into the category of time-and-location based protocols which we will introduce in the next parts. However, it does not actually use the timing data directly. It assumes that the sender sends its location (with the maximum relative error equal to δ) along with the timestamp showing the transmission time (with a maximum relative error of Δ). If the nodes move with a maximum speed of v, then this method gives an upper bound on the distance which is found by $d \leq \left|\left|loc_s - loc_r\right|\right| + 2v(t_r - t_S) + \delta$.

The authors also proposed a time-based approach which they called "temporal leashes". In this approach the transmission time is (authentically) written in the packet and the receiver

can then decide on the distance from the duration of signal flight. To authenticate the timestamp (and location in the previous part), digital signatures and the hash-chain-based (tree) authentication methods have been used. However the original idea comes from TESLA authentication method (Perrig et al., 2000).
Regardless of the protocol types, for both parties to be able to measure the distance, each must have the chance to challenge the other. So, at least three messages must be exchanged. For unidirectional neighbor discovery, exchanging two messages suffices (Sayad et al.,2008). Korkmaz addressed the RTT measurement, focusing on the difference of signal propagation speed in the healthy nodes and adversaries' channels (Korkmaz, 2005). In this method, a combination of power and delay-related criteria has been used. The simplified diagram of Korkmaz protocol is depicted in Figure 3. Here, "M" is an authenticated request message and "ACK" is its reply containing P_{r_m}, P_{t_a} and $t'_{t_a} - t'_{r_m}$ which is the processing delay in B.

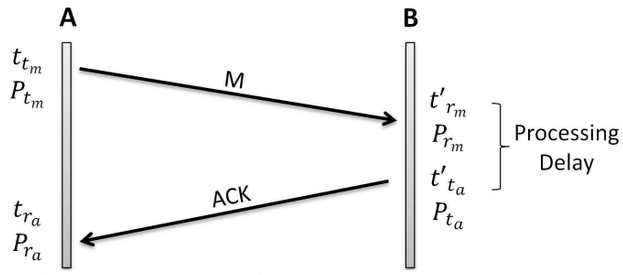


Fig. 3. Korkmaz's neighbor discovery protocol

If the worst case signal propagation speed between two nodes with distance R is s, and the maximum speed of signal propagation in the adversaries' media is v_{adv} (v_{adv}>s), then the upper and lower bound estimates of the RTT are found as $[\frac{2R}{v_{adv}}, \frac{2R}{s}]$. Any claiming neighbor with measured RTT lower than $\frac{2R}{v_{adv}}$ is accepted while anyone with a RTT larger than $\frac{2R}{s}$ is rejected. Not all the measurements which fall in this interval are accepted. If the actual speed of signal propagation in the network is x, then Korkmaz proposes using a hard decision making threshold for the RTT samples falling in the abovementioned interval. The following formula normalizes the measured RTT.

$$\varepsilon_c = \frac{\frac{2R}{s} - RTT}{\frac{2R}{s} - \frac{2R}{v_{adv}}} \tag{1}$$

The hard decision making threshold is the acceptable level of confidence i.e. for $\varepsilon_c \geq L_c$ we assume that the two nodes are neighbors and for $\varepsilon_c \leq L_c$ we assume they are not. This threshold is equivalent to an effective distance $R_{eff} = x(1/s - L_c(1/s - 1/c))R$. Obviously, this distance can be either smaller or larger than R. So this type of decision making is always prone to either rejecting correct neighbors or accepting some incorrect ones which opens a vulnerability window for the system. To mitigate this flaw, Korkmaz proposed to combine the RTT-measurement-based method with a power-measurement-based one. Generally, the relationship between the distance, transmitted and received signal strengths is:

$$d = k(\frac{P_t}{P_r})^{n^{-1}} \tag{2}$$

where k and n are constants and are determined by the characteristics of channel and communicating devices. Since the ACK message contains the measured power fields, A can easily compute the distance (d) using equation (2). However, since k and n fluctuate in the real world scenarios it is proposed to verify the following equilibrium instead:

$$\frac{P_{t_m}}{P_{r_m}} = \frac{P_{t_a}}{P_{r_a}} \tag{3}$$

Korkmaz claims that if the nodes are not actual neighbors i.e. they communicate through relaying adversaries, then the values of P_{r_m} and P_{r_a} will be altered depending on the transmission power used during relaying and the distance between the adversaries and legitimate nodes, and hence, the above equilibrium does not hold. However, it can be easily verified that if the relaying adversaries collaborate with each other, this countermeasure is easily neutralized. For example in Figure 1, if C tells D to send the signal received from A, to B with the same power she received (and vice versa), then eq. (2) is satisfied. Also it should be emphasized that this method only addresses probabilistic security facing external attacks. Obviously B, as an internal attacker can deceive A if she lies about the processing time.

As an alternative approach, a few authors suggested searching for graph abnormalities to detect the relaying attackers either in a distributed or centralized manner. Maheshwari et al. proposed a distributed geometrical algorithm to search for graph abnormalities in wireless networks (Maheshwari et al., 2007). The core idea of their work was to find impossible cases which do not occur in healthy graphs. If two nodes are claimed not to be neighbors then the number of their independent neighbors (those neighbors of the two nodes who cannot see each other) is quite limited. For example consider Figure 4 in which two healthy nodes a and b are at their farthest possible range of neighborhood. In this case, the nodes which are considered to be neighbors of both of them can only be found in the intersection of the two coverage areas (the hatched area). Mashewari et al. have proved that no more than two independent neighbors can be found in the hatched area in this case. So if a third common neighbor comes in, then it must be covered by one of the previous common neighbors. The number of independent common neighbors decreases even more when the two nodes further move away from each other. So if the adversaries launch a wormhole attack like the one in Figure 5, then two independent nodes a and b which are far from each other, will have three common independent neighbors e, f and g and according to the argument we made there can be no more than two common independent neighbors in this case and hence, a fraud has been detected.

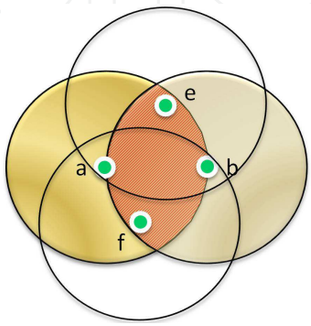


Fig. 4. Mashewari et al.'s diagram to demonstrate the maximum number of independent common neighbors of two nearly independent nodes

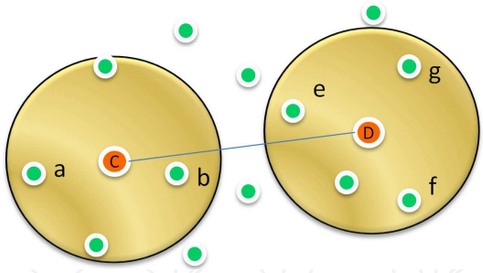


Fig. 5. A sample wormhole attack which is detected by Mashewari et al.'s forbidden structure search

However this was a simple example and the authors have extended this one-hop graph-based idea to a k-hop one to obtain a higher wormhole attack detection probability.

In a combined work of distance measurement and graph abnormality detection, Shokri et al. proposed what they called a practical secure neighbor verification protocol for sensor networks (Shokri et al., 2008). The core idea of their measurement part contribution relies on having two different transceivers (one Radio Frequency (RF) and one Ultra Sonic (US)) which is a bit similar to the echo protocol approach (Sastry et al., 2003). Since the sensors have simple processors of microsecond clock pricision, the RF transceiver is used for clock synchronization-like purposes (or simply measuring the difference of time). Distance measurement with RF signals requires a nanosecond clock precision. So the RTT measurement (and consequently the distance measurement) is done more accurately using the slow-propagating US signals with the precision at hand.

This method is actually composed of three smaller sub-protocols. Assuming that all pairs of the nodes already have a symmetric shared key (like K_{AB} between A and B), each initiating node A, sends a request message to (arbitrary) node B over the RF channel. B replies to this request with another message. Then A broadcasts a message through its ultrasonic transmitter to reveal its nonce whose hash was previously sent at the request transmission phase. This is similar to the delayed authentication methods (like TESLA method (Perrig et al., 2000)) which bind the authenticity of the next step to the previous one. At last an ACK is sent to each of the candidates (like B). The whole message exchange of sub-protocol 1 is depicted in Figure 6. Notice that $E_K(.)$ denotes an encryption with key K, and N_{AB} and N_A are two nonces. $MAC_K\{.\}$ means that the whole message is protected with a message authentication code under key K. Node B's side times are measured with its own clock which is not necessarily synchronized with A. At the end of this phase, B can compute its distance to A using $\hat{d}_{AB} = s.[t'_{rng,B} - t_{rng,A} - (t'_{req,B} - t_{req,A})]$ where s is the speed of slowly propagating ultrasonic signal.

After this stage, the nodes exchange their local table of distances as the second step of the protocol so that each obtains a local view of the network topology. At the third stage, each node starts to run a series of tests on the information obtained in the previous stages. First of all, a node eliminates those links with distances outside the acceptable range. Second, the symmetry of the links is verified meaning that an arbitrary node A checks the $d_{AB} = d_{BA}$ equaility for any candidate B. Third, for any claiming neighbor, the testing node tries to find two other candidates in the table and then, knowing the distances between them, it assumes hypothetical (but wisely selected) positions for these three candidates so that the triangular inequalities hold for each possible set of three nodes (including itself). At last, it verifies

whether the four nodes form a convex quadrilateral[3] or not. The authors have shown that the above criteria, prevent wormhole attacks with two colluding adversaries and also resist well facing wormhole attacks with more than two adversaries.

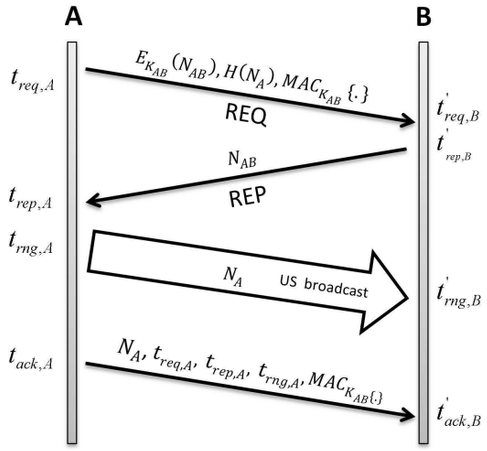


Fig. 6. The sub-protocol 1 (distance measurement) of Shokri et al.'s method

Although Shokri's protocol has novel contributions and combined many ideas like delayed authentication and geometrical tests, it suffers from some design flaws. In an infrastructureless wireless network, neighbor discovery is the first protocol which is run i.e. there is no information about the existence of the nearby nodes prior to starting the protocol. So there are no agreed mutual cryptographic keys and hence *A* cannot send authenticated messages to a node like *B* which is not even known to be in its vicinity. Sensor nodes are extremely resource-limited devices and in a network with thousands of sensors, it is practically impossible to store a large number of symmetric keys in a node's memory and this has been a challenging security problem thus far (Chan et al., 2003)(Du et al., 2005)(Zhu et al., 2003 & 2006). Using LEAP (Zhu et al., 2003) or LEAP+-like (Zhu et al., 2006) key distribution methods selected by the authors, is problematic since these protocols themselves have neighbor discovery protocols in their initial setup phase and static networks do not need multiple-time independent neighbor discoveries. Besides, LEAP+ itself is capable of blocking wormhole attacks after the key establishment phase. With a rather high number of message transmissions (which includes the hidden stage of table exchanges) and a high number of link verification computations, this solution is not very energy conserving. Also the use of two transceivers (RF and US) is in contradiction to the basic assumption in sensor networks design which is cost effectiveness. It should be mentioned that this protocol is unidirectional in our terminology and is designed to prevent external wormhole attacks in static networks, thus has limited functionality facing internal attackers and also in highly mobile networks. However, compared to the older centralized approaches for graph abnormality detection (Rasmussen et al., 2007), both of the abovementioned distributed approaches are valuable.

[3]A convex quadrilateral with four vertexes A, B, C and D is characterized by $\left(\overrightarrow{AB} \times \overrightarrow{BC}\right)\left(\overrightarrow{BC} \times \overrightarrow{CD}\right)\left(\overrightarrow{CD} \times \overrightarrow{DA}\right)\left(\overrightarrow{DA} \times \overrightarrow{AB}\right) > 0$

Poturalski et al., in a series of evolutionary papers and technical reports, tried to classify neighbor discovery protocols into two generic time-based and time-and-location-based classes and provide systematic rules to formally verify the security of these protocols against external attacks (Poturalski et al., 2007) (Poturalski et al., 2008). In their initial technical report, they formally derived a so-called impossibility result for the time-based neighbor discovery protocols stating that it is impossible for a neighbor discovery protocol which solely relies on signal propagation time measurements to provide seamless security (Poturalski et al., 2007). Figure 7 demonstrates the "impossibility result" informally. If the maximum allowable distance of two healthy neighbors is *r*, then the maximum signal travel time would be *r/s* where *s* is the speed of signal propagation in the channel. If two nodes cannot reach each other directly, either due to a large distance (Figure 7c) or a barrier (Figure 7b) then one or more relaying adversaries can deliver the messages and as long as the time of signal flight is less than *r/s,* this relay will not be detected. However, if the imposed delay of a relay is more than *r/s* (i.e. Δ_{relay}>*r/s*) then time-based protocols can become secure. It is also shown that designing secure time-and-location based protocols is possible since the receiving nodes have the sender's location (in an authentic manner carried by the message) and compute the valid distance themselves and then compare it with the one obtained from time measurements to detect probable attacks.

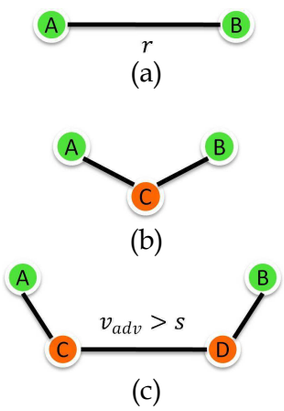


Fig. 7. There is no possibility for B (or A) to always distinguish between these three paths in a propagation time measuring protocol as long as the propagation delay does not exceed *r/s*. v_{adv} is the signal propagation speed in the adversaries' channel which is higher than *s*.

Later, each of the two classes of protocols proposed by Poturalski et al., was divided into two groups; Beacon (B) and Challenge/Response (CR) protocols (Poturalski et al., 2008). In B-protocols, a node broadcasts some information without having the response of the other side. The receivers are supposed to add the sender to their list of neighbors after some processing and hence, this method is unidirectional. On the other side, CR-protocols support the minimum three-phase message exchange requirement for a bidirectional neighbor discovery. However the samples the authors have mentioned for the CR protocols are not really bidirectional (although they could be as in their original framework). In Poturalski's examples, CR protocols create the chance of challenge for one side only, and all the other nodes must run the same protocol themselves to complete the neighbor discovery task.

The authors claimed a secure neighbor discovery protocol is characterized with two properties: "correctness" i.e. if the protocol declares two nodes neighbors at some time, they must indeed be neighbors at that time, and "availability" meaning that if two nodes remain neighbors for some time (T_P) then the protocol must detect this neighborship.
They defined seven types of events to describe their rules and protocols with. Table 1 summarizes six of them which we will deal with, along with their description.

Receive(A;t;m)	*A* receives the first bit of message *m* at *t*
Bcast(A;t;m)	*A* broadcasts message *m* at *t*
Fresh(A;t;n)	Nonce *n* is freshly generated by *A* at *t*
Neighbor(A;t;B;C;t')	At *t*, *A* declares *B* has been a neighbor of *C* at *t'*(unidirectional)
NDstart(A;t)	*A* starts a neighbor discovery with all the nodes at *t*
NDstart(A;t;B)	*A* starts a neighbor discovery with *B* at *t*

Table 1. Some of the events symbols and their description in Poturalski et al.'s literature.

Based on these definitions, they formally defined security requirements for the four possible groups of neighbor discovery protocols and presented a sample pseudo-code for each group satisfying those set of requirements. To unify the demonstrations, we have converted the codes to message diagrams. Figure 8 and Figure 9 show Poturalski's sample protocols for beacon time-based (B/T) and beacon time and location-based (B/TL) neighbor discovery protocols respectively. In Figure 10 and Figure 11, challenge-response versions of these protocols are depicted. Notice that some of the messages in these figures are actually intended to be received by a single node. However to comply with the broadcast-type message transmission symbol shown in Table 1 (and in the pseudo codes), they are drawn with broadcast-like arrows.
In all of these figures, len{.} stands for an operator giving the length of a message transmission (in seconds for example), *r* is a node regular transmission range and *s* is the signal propagation speed in the network communication channel.

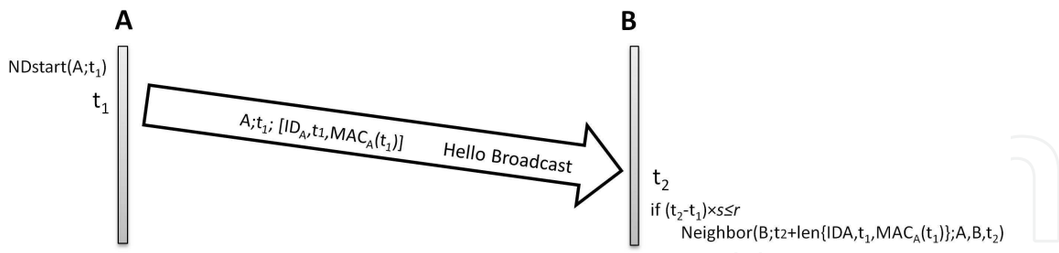


Fig. 8. A beacon and time-based protocol pseudo code (B/T)

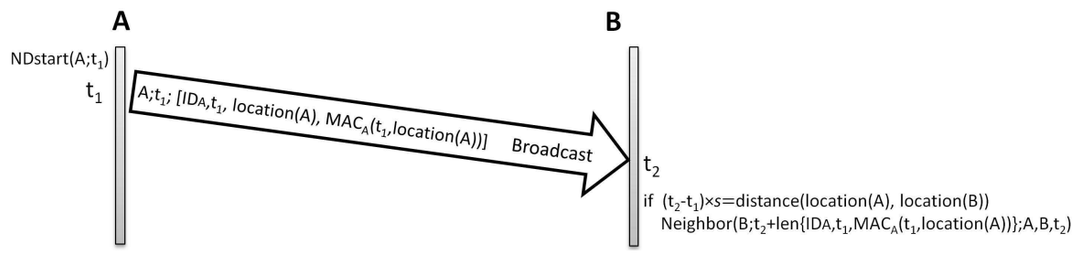


Fig. 9. A beacon and time-and-location-based protocol pseudo code (B/TL)

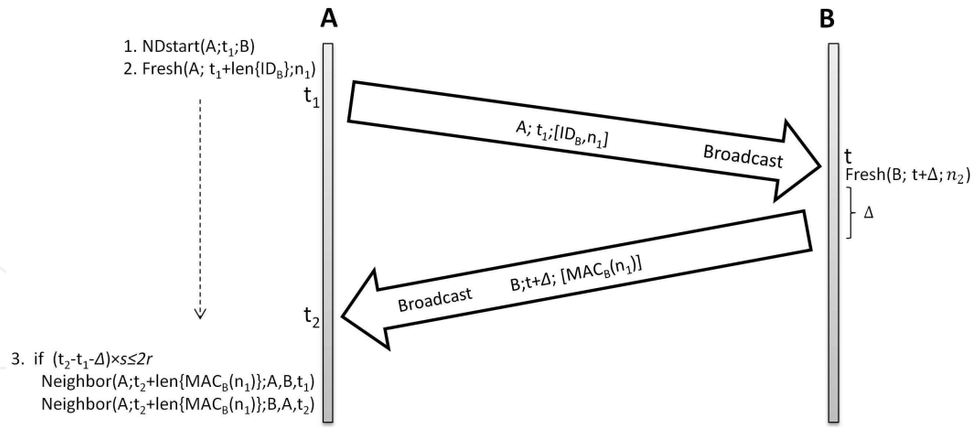


Fig. 10. A challenge-response time-based protocol pseudo code (CR/T)

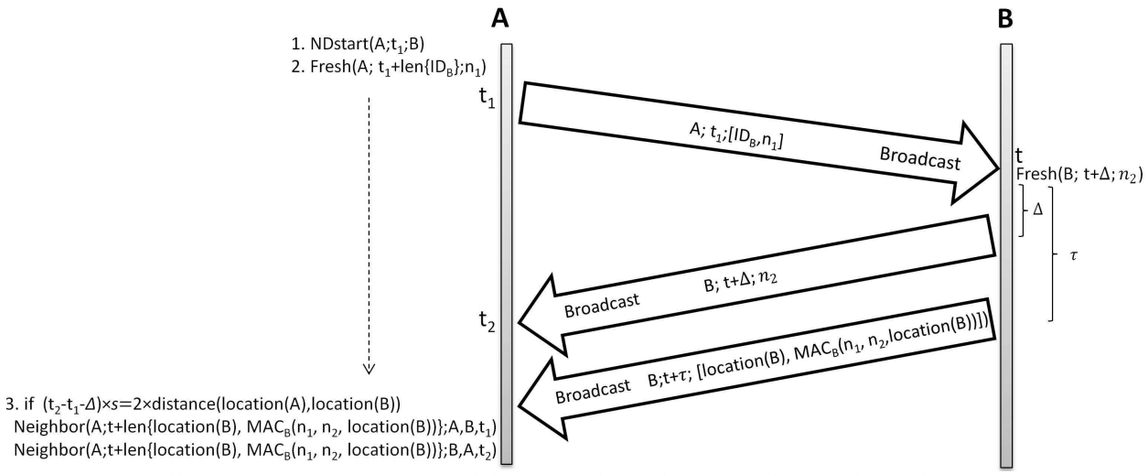


Fig. 11. A challenge-response time-and-location-based protocol pseudo code (CR/TL)

As we described in the previous parts, if there are no challenges and responses the adversaries can easily relay the broadcasted beacon packets to other parts of the network. Also notice that the protocols times are written from a third-party's point of view. In practice, *B* and *A* have time synchronization problems too, which even widens the vulnerability window more in beacon-based protocols.

In all of these sample protocols, nodes need to verify the authentication codes attached to the end of messages which in turn makes the protocols dependent on another key distribution protocol (involving either storage of all the keys in the memory of the nodes that makes it only suitable for ad hoc networks or employment of a key agreement protocol which inherently needs a neighbor discovery protocol itself).

Regardless of these practical issues, each of these simplified representatives of neighbor discovery protocols has been proven to satisfy Poturalski's mathematical security requirements under some specific conditions. Beacon time-based (B/T) protocols are secure if the adversary's relaying delay is greater than or equal to $r.s^{-1}$. This is a rather obvious constraint since with less than this bound, the adversary can relay some messages without adding too much delay and keep the overall propagation and relaying delays below the acceptable threshold $r.s^{-1}$ (refer to Figure 7). It can also be verified that the availability

property is also satisfied with $T_{pB/T} = \text{len}\{ID_A, t, MAC_A(t))\} + r.s^{-1}$ i.e. if two nodes remain neighbors for this amount of time (and the B/T neighbor discovery is started) then, this neighborship will be definitely detected by the protocol. Similarly, for a TL protocol to be secure, the designer must take a communication media whose signal propagation speed is close to the maximum possible speed in an adversary's channel (or simply the light speed (s=c=v_{adv})). If less than this speed is used (i.e. v_{adv}>s), even though A has both B's and its own locations, a pair of adversaries can simply launch the wormhole attack keeping the overall RTT equal to 2×distance(A,B)/s which is sufficient to satisfy the protocol criterion. The security conditions of other protocols can be found similarly. Table 2 summarizes the conditions under which each of the four groups of neighbor discovery protocols is secure.

Protocol Type	Security Constraints Applied
B/T	1. $\Delta_{relay} \geq r.s^{-1}$ 2. $T_{pB/T}$=len{ID_A,t,MAC_A(t)}+ $r.s^{-1}$
CR/T	1. $\Delta_{relay} \geq 2r.s^{-1}$ 2. $T_{pCR/T}$=len{ID_B,n}+len{MAC_B(n)}+ $2r.s^{-1}$
B/TL	1. $\Delta_{relay} \geq 0$ 2. $s = v_{adv}$ 3. $T_{pCR/TL}$=∞ (depending on distance)
CR/TL	

Table 2. Conditions for each class of neighbor discovery protocols to support Poturalski's formal description of security requirements

We investigated some of the main external-attack-related researches and the positive and negative aspects of them in this section. In the next section we focus on the internal attacks' countermeasures which are different from the previous methods in nature and mostly are cryptographic solutions.

3. Neighbor Discovery Internal Attacks Countermeasures

There are a few internal attacks affecting the performance of neighbor discovery protocols. Needless to say some of the external attacks have the internal form too. For example, in a wormhole attack, two internal relaying adversaries can even alternate the packet contents in order to deceive healthy entities. Unfortunately compared to the external attacks described in the previous section, the number of these attacks is not limited at all. This is because these attacks are more related to the protocol-specific features than the nature of neighbor discovery. As there are many variants for each neighbor discovery protocol family, the internal attacks are also numerous. However, there are a few common attacks that can conceptually cover some of these threats, and, among them, broadcast attacks are more outstanding.

Almost all the algorithms which use a broadcasted data are susceptible to being invaded by the broadcast-type attackers. In the sensor networks hello flooding case, a loud advertisement can convince many surrounding nodes that the adversary is one of their neighbors. This attack can be launched in the routing algorithms too e.g. the adversary can rebroadcast the received RREQ packet with high power to be a part of the best routes to the destination with a high probability. Notice that unlike the previous part, in internal attacks,

the adversary is a legitimate node i.e. a node is captured and compromised by an adversary and all of its cryptographic keys and private information are known to the adversaries.

Broadcast attacks had been previously addressed with different names somewhat but Karlof and Wagner were the first who introduced this attack specially in sensor networks (Karlof et al., 2003). Consider a simple form of neighbor discovery in which a well-equipped internal (legitimate) adversary tries to broadcast a packet called hello (beacon) and every node that hears this packet adds the sender to its neighbors list. Since one side of the protocol which sends the authenticated messages is adversary herself, it is impossible to rely on either time or location information given by the other party. So generally to defeat an internal attacker every node must rely on its own data. The same argument holds true for the CR protocols. If the adversary gives false (but authenticated) location information to the other party, then colluding with other relaying adversaries she can easily bypass the normal security checkpoints.

As a solution for the broadcast-type attacks, Karlof et al. proposed to verify bidirectionality of the links (Karlof et al., 2003). They assumed that the adversary has a high-power transmitter but an ordinary receiver and thus is unable to capture distant weak signals. If the receiver (*B*) sends valuable information in reply to the broadcasted message (hello), on which the adversary (*A*) must rely for future communications, then she will be defeated since she cannot hear *B*. The initial raw idea of verifying bidirectionality of the links was not developed much by the authors, however, it was mentioned that this countermeasure is useless if the adversary has both a high-power transmitter and a sensitive receiver.

In sensor networks, some authors then tried to limit the nodes communication ranges through cryptographic methods (Du et al., 2005)(Lin et al., 2005)(Zhu et al., 2006). The early methods, tried to pre-load a large number of pairwise keys in each node's memory (key pre-distribution). But for sensor networks, this was an impractical solution since the amount of memory each sensor has is quite limited. Probabilistic key pre-distribution schemes were proposed to solve this problem. In these methods, after the deployment, every node tries to find some common keys with its neighbors through a so-called mutual key discovery protocol (Eschenauer et al., 2002). However these solutions had problems too. The common key discovery was itself another protocol which was needed to be secured. Besides probabilistic approaches do not always guarantee providing a common key. So another approach was adopted which was letting the nodes themselves establish the keys after deployment (Lin et al., 2005)(Zhu et al., 2006). This implies the use of a negotiation protocol between the nodes when they are deployed. It is rather obvious that to protect these negotiations against external attackers the messages must be encrypted with some key and since the mutual keys are not known at this phase yet, usually a pre-loaded global key is used.

A general assumption made by these methods is that compromising a node takes time but to stop an attacker who can capture and compromise the nodes, the whole negotiation should not take more than T_{min} seconds. To prevent further attacks, nodes themselves delete the global key from their memory after T_{min} seconds.

Secure Cell Relay (SCR) is one of the distributed key establishment/routing protocols which resists broadcast attacks (Lin et al. 2005)(Du et al., 2006). It uses a three-way handshake protocol to avoid the unidirectional link problem. There are two versions of this protocol but the most recent one in which the location information has also been used is depicted in

Figure 12. In this figure, K is the global shared key, E_K the encrypting operator, and N_0 is a nonce. K_B^b is defined as the B's broadcast key and K_{AB} is the private key between A and B used for later communications. At the end of this process, B adds A to its neighbors list and stores K_{AB} and K_B^b in a table for the future. After completion of the protocol for every node, all the nodes delete the global key form their memory.

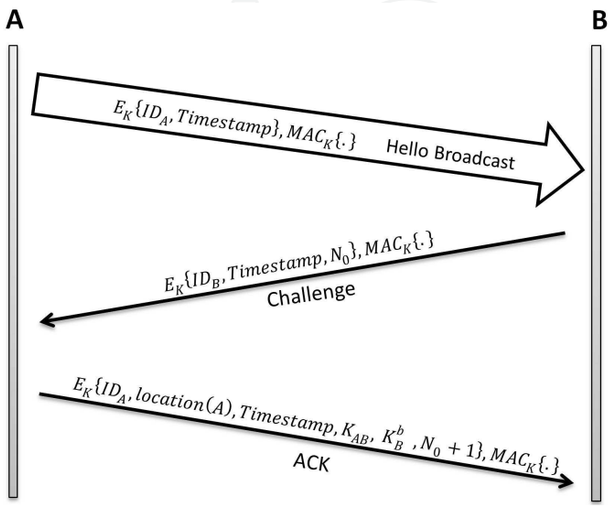


Fig. 12. Secure Cell Relay (SCR) neighbor discovery protocol message diagram

But SCR neighbor discovery protocol is weak in many aspects. The use of time stamps forces the designer to somehow maintain a synchronized clock which is problematic in distributed networks. Besides with a good design, in a three-phase message exchange protocol, both nodes can add each other to their neighbors list because both of the parties had the chance to challenge the other one. However, SCR messages need to be modified to provide this feature. Also notice that after erasing the global key K, there is no chance for a node to update its neighbors list. This property makes the protocol unsuitable for mobile scenarios.

LEAP (Zhu et al., 2003) and then LEAP+ (Zhu et al., 2006) are key distribution protocols for sensor networks which also block the broadcasting adversary. The main goal of LEAP+ was to create four sets of keys for each node; one set of pairwise keys for inter-neighbor communications, one key for local broadcasting, one globally-shared network key and one key to communicate with the sink. As the local broadcast key is made from the local pairwise keys and dealing with the attack the two others are not needed, we only focus on the construction of pairwise keys.

Here again the assumption of an adversary-free immune network after the initial deployment is necessary. So we assume that for an interval which is at least T_{min} seconds there is no internal attacker present in the network. To prevent external attacks during this period, there is a globally shared key pre-loaded into memory of the nodes.

If f_k is a one-way keyed pseudo-random function (like encryption operators) with key k, then the pairwise key construction in LEAP+ can be summarized as shown in Figure 13.

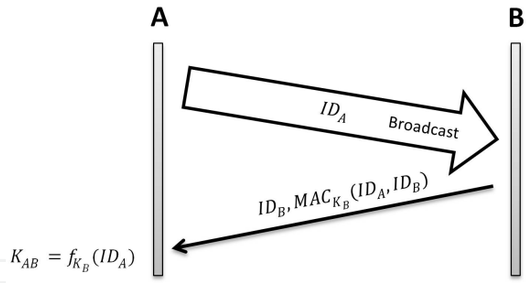


Fig. 13. The simplified form of key establishment in LEAP+ protocol at the initial deployment phase

"*A*" finds the pairwise key by computing $K_{AB} = f_{K_B}(ID_A)$ where K_B is found by applying the global key to the one-way function with *B*'s ID as the input argument ($K_B = f_K(ID_B)$). Once this process is done for every node, the global key is erased from the memory along with K_B. After the pairwise key establishment phase, all the communications are done in an encrypted manner with the previously generated keys. Since the adversary is supposed to capture a node not sooner than T_{min} seconds, she is unable to send meaningful messages to nodes farther than the communication range of a normal node. So, broadcast attacks are thwarted this way. Also, the adversary cannot make new pairwise keys anymore since the required global key is missing in the memory of the captured node.

Although this protocol performs well in terms of resource consumption and complexity, it has drawbacks too. A closer look at the protocol reveals that *B*'s presence in this challenge-response-like protocol is not as strong as it should be. The only role of *B* is announcing its presence by sending back its ID. It does not even generate any random number to maintain the freshness of the key. Although it has been assumed that the network is devoid of any internal adversaries for at least T_{min} seconds (since for example it takes at least T_{min} seconds to intrude into a tamper-resistant device memory), this is not a valid assumption for the external ones. If external adversaries are present before the nodes deployment, they are able to launch relay-based attacks which simply means, in the above protocol, *B* could be a relayed distant node. After T_{min} seconds, the adversary intrudes into *A*'s memory and at the end, she has a large-range communication capability for which she had planned before. It is also obvious that due to the erasure of the global key after T_{min} seconds, nodes are unable to restart the neighbor discovery protocol in the future and thus, this protocol does not support mobility. Any mobility-supporting security framework must allow periodical updates of the neighbors list. This implicitly involves participation of either time or a random number (nonce) in the protocol to block replay attacks. The authors have virtually limited the range one node can communicate in, through cryptographic methods. With the erasure of global key (and the static network the authors assumed) the probable future neighbor discoveries are limited to the detection of lost connections due to power depletions or failures.

In (Sayad et al., 2008), as an alternative solution, especially in sensor networks in which broadcast attacks are more devastating, we proposed a probabilistic robust design framework for the internal broadcast attacks which has a very low complexity and can even be combined with any other secure neighbor discovery protocol. To define a probabilistic robustness, we shall first differentiate three general attack profiles.

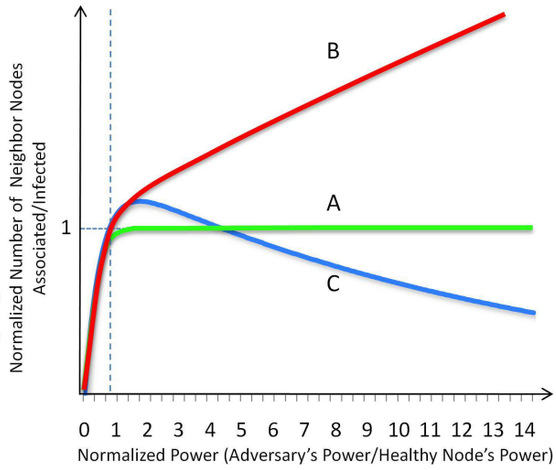


Fig. 14. (A) An optimal secure neighbor discovery protocol broadcast attack profile (B) An unprotected neighbor discovery protocol broadcast attack profile (C) A broadcast attack-resilient neighbor discovery profile

As shown in Figure 14., facing broadcast-type attacks, the behavior of neighbor discovery protocols can be categorized into three different profiles. When the transmission power of a node or sensor increases, the number of neighbors is also increased (conforming with the different transmission ranges nodes have based on their available power resources), but if the power of a node goes beyond a threshold corresponding to the maximum allowable transmission range, then these three family of protocols behave differently. Regarding the broadcast attacks, type A profile is the desired optimal secure neighbor discovery protocol i.e. if an adversary tries to increase her power in order to obtain more neighbors, she will not succeed. Complex cryptographic solutions may be put in this group.

Type B profile shows the performance of an unprotected neighbor discovery protocol. The adversary's payoff increases as its power goes up. In type C which is actually the focus of our work, the maximum payoff (possible number of neighbors) of the adversary is not as low as an optimal secure solution but definitely bounded. In this approach, the designer tries to simplify the protocol in expense of losing some (but not all) of the security. This is a general framework however and the amount of penalty paid for simplicity depends on the designer's approach. It is worth mentioning that the initial parts of all these three profiles are similar, because as long as an adversary's behavior falls in the class of healthy nodes' behavior, she receives the same amount of payoff as a normal node does. So, there is no instant solution which can decrease the payoff of an internal adversary to less than that of a healthy node.

We propose that one can wisely manipulate Medium Access Control (MAC) protocol parameters to achieve a type-C resistivity against broadcast attacks in sensor networks without consuming too much energy. In a compromised network, when an adversary that is equipped with both powerful transmitter and sensitive receiver, broadcasts a hello-like request or beacon (hello flooding), a lot of nodes receive it almost simultaneously. In a two (or more) way handshake protocol, these nodes will start to compete to grab the channel and send the reply messages in order to announce their presence. Healthy nodes have small transmission and reception ranges. Therefore, roughly speaking, those nodes located farther than the carrier sense range of each other will try to send the reply messages back almost

simultaneously. The main idea is to tune the channel access and transmission parameters so that the responses of these distant nodes collide with each other at the adversary's receiver due to the high density in arrival. If these reply messages contain valuable information which is vital for future communications, then as the adversary is unable to decode the reply messages correctly, she is obliged to reduce her power. This is somehow similar to the well-known hidden node effect in wireless ad hoc networks. Figure 15. shows the attacker and sample distant nodes colonies along with their corresponding transmission ranges.

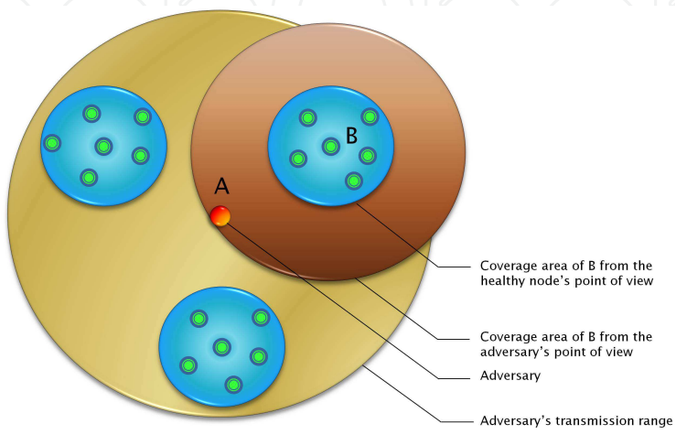


Fig. 15. Transmission ranges of normal and adversary nodes in Sayad et al.'s framework. Adversary's high transmission power attracts many healthy nodes. However, far healthy nodes do not see each other and hence start to transmit reply packets simultaneously

As in the neighbor discovery phase no infrastructure is already formed, a random channel accessing method is preferred. So one can expect that this approach would be a probabilistic one at the end. Although the design framework introduced is quite general, we give an instance protocol in which the notion of important information delivery in the reply packets transmission phase is contrived. The sample protocol in Figure 16 tries to establish a mutual key between the two nodes using Diffie-Hellman key agreement protocol (Diffie & Hellman, 1976) while following the framework rules.

It should be noticed that neighbor discovery message collisions must be minimized in a healthy scenario i.e. since the only difference between healthy and adversary nodes is their transmission ranges, high-colluding protocol design for the adversary might also affect the normal behavior of the network and this is to be avoided. In our example, this leads to a trade-off problem.

To make a demonstration of the above proposal's efficiency, for channel accessing, IEEE 802.15.4 WPAN standard was chosen (IEEE 802.15.4, 2006). However, some modifications were made to the protocol to adapt its neighbor discovery to the classic three-phase type, and the minimum and maximum back-off exponents were considered as the parameters to be tuned. With a node distribution density of 0.01 (1 node in $100m^2$), the attack profile for different values of the minimum back-off exponent is shown in Figure 17.

As it can be clearly seen, this method behaves like a type-C attack-resistant protocol. With the maximum back-off exponent set to eight and the identical reply packets which were seven back-off periods long, the minimum back-off exponent was swept. With this configuration, the adversary's payoff increases as the minimum back-off exponent goes up.

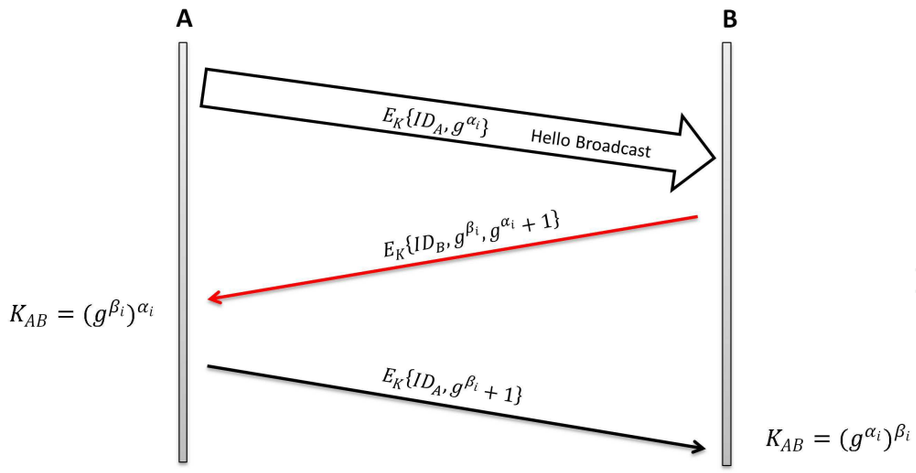


Fig. 16. An example of Sayad et al.'s framework. Reply message includes a part of the common key which will be used for future communications

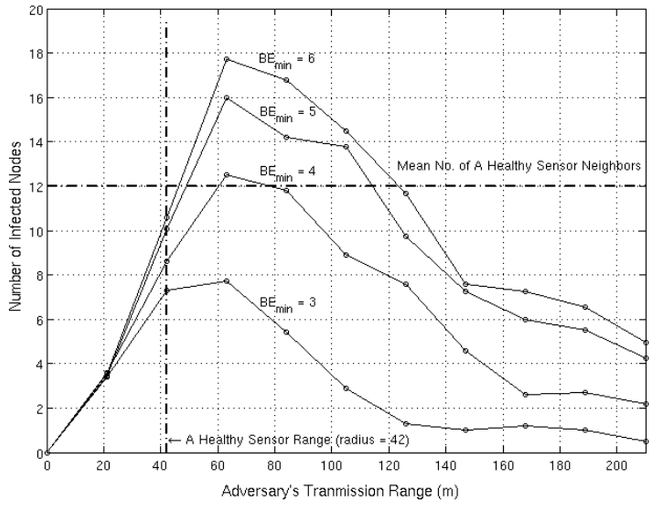


Fig. 17. The broadcast attack profiles on Sayad et al.'s sample protocol. The adversary receives a negative feedback from the network in response to her greediness in obtaining more neighbors

Notice that we also have to keep the healthy neighbor discovery scenario as intact as possible i.e. any profile closer to the intersection of the mean number of a healthy node's neighbors line and the normal transmission range line is preferred. With a high minimum back-off exponent, the profile approaches the ideal point for healthy scenarios, however, the maximum (but still restricted) payoff of the adversary is also increased. Now it is up to the network designer to set how much security (with respect to the broadcast attacks) should be sacrificed for sake of gaining more efficiency.

The abovementioned design framework is a concept which can be freely combined with the other neighbor discovery protocols and since it has a very low cost, is quite applicable in resource-limited sensor networks. Unlike the cryptographic methods (and many other

described protocols), this framework supports (high) mobility of the nodes. Besides, in this approach, the limiting assumption about the initial adversary-free period of the network operation at the time of nodes deployment has been removed.

4. Conclusion

In this chapter, we introduced many of the current threats and solutions for the neighbor discovery problem in wireless ad hoc and sensor networks. Separating the attacks into internal and external ones, depending on how much the protocols are resistant to each of these attacks, we categorized them into two groups as well.
Many of the current solutions focus on the external relay-based attacks while little effort has been made to mitigate the internal attackers' damage. Most of the current solutions for the internal attacks are cryptographic which are either resource consuming or do not support mobility.
Numerating all the solutions, we proposed a probabilistic countermeasure for the internal broadcast attacks. It is obvious that combinational attacks require combinational solutions. Our design framework for the internal broadcast attacks can be combined with nearly all the other solutions to provide higher levels of security facing both internal and external attacks.

Acknowledgement

This work has been financially supported by Iran Telecommunication Research Center (ITRC). The authors also wish to thank Prof. Vijay Varadharajan form Macquarie University and Alireza Mohammadi-nodooshan from K.N.Toosi University of Technology for their scientific support.

5. References

Akyildiz, F.; Ian, W.S., Sankarasubramaniam, Y. and Cayirci, E. (2002). A survey on sensor networks, *IEEE Communication Magazine,* vol. 40, Aug. 2002., pp. 102-114

Brands, S.; Chaum, D. (1993). Distance-bounding protocols, *Proceedings of Eurocrypt '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pp. 344-360, May 1993, Springer

Chan, H. ; Perrig, A. & Song, D. (2003). *Random key predistribution schemes for sensor networks*, *Proceedings of IEEE Security and Privacy Symposium*, pp. 197-213, 2003

Diffie, W.; Hellman, M.E. (1976). New directions in cryptography, *IEEE Transactions on Information Theory,* vol. 22, No.6, Nov. 1976, pp. 644-654

Djenouri, D.; Khelladi, L., Jamel, D. and Badache, N. (2005). A survey of security issues in wireless ad hoc and sensor networks, *IEEE Communication Surveys,* vol. 7, 4th quarter 2005, pp. 2-28

Du, W.; Deng, J., Han, W.S., Varshney, P.K., Katz, J., Khalili, A. (2005). *A pairwise key predistribution scheme for wireless sensor networks.* ACM Transactions on Information and System Security (TISSEC), 2005 , pp. 228-258, 2005, ACM Press

Du, X. ; Xiao, Y., Chen, H.H. (2006). Secure cell relay routing protocol for sensor networks: Research Articles, Vol. 6, No. 3, *Wireless Communications and Mobile Computing*, May 2006, pp. 375-391

Eschenauer, L. ; Gligor, V. (2002). A Key Management Scheme for Distributed Sensor Networks, *Proceedings of ACM Conference on Computer and Communications Security*, pp. 41-47, USA, Apr. 2002, Washington DC

Hu, Y.C; Perrig, A., Johnson, D.B. (2006). Wormhole attacks in wireless networks, *IEEE Journal on Selected Areas in Communications,* Vol. 24, No. 2, Feb. 2006, pp. 370-380

IEEE 802.15.4 Standard (2006). Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), *IEEE*, Sep. 2006

Karlof, C. ; Wagner, D. (2003). Secure Routing in Sensor Networks: Attacks and Countermeasures, *Elsevier Ad hoc Networks,* Vol. 1, No. 2-3, Sep. 2003, *pp. 293-315*

Korkmaz, T. (2005). Verifying Physical Presence of Neighbors against Replay-based Attacks in Wireless Networks, *Intenational Journal of Information Technoilogy,* Vol. 11, No. 2, 2005., pp. 9-20

Lin, F. ; Du, X. (2005). Secure cell relay routing protocol for sensor networks, *Proceedings of IEEE Computing and Communications Conference*, pp. 477-482, Apr. 2005

Maheshwari, R.; Gao, J. & Das, S.R. (2007). Detecting wormhole attacks in wireless networks using connectivity information, *Proceedings of IEEE International Conference on Computer Communications*, pp. 107-115, May 2007

Papadimitratos, P.; Poturalski, M., Schaller, P., Lafourcade, P., Basin, D., Capkun, S., Hubaux, J.P., (2008). Secure neighborhood discovery: a fundamental element for mobile ad hoc networking, *IEEE Communication Magazine,* vol. 46, No. 2, Aug. 2002., pp. 102-114

Perrig, A.; Canetti, R., Song, D., Tygar, D. (2000). Efficient Authentication and Signing of Multicast Streams over Lossy Channels, *Proceedings of IEEE Symposium on Security and Privacy*, May 2000

Pickholtz, R.L.; Schilling, D.L. & Milstein, B. (1982). Theory of spread spectrum communications: a tutorial, *IEEE Transactions on Communications,* Vol. 20, No. 5, May 1982, pp. 855-884

Poturalski, M.; Papadimitratos, P., Hubaux, J.P. (2008). Towards provable secure neighbor discovery in wireless networks, *LCA Report (EPFL University)*, Oct. 2008

Poturalski, M.; Papadimitratos, P., Hubaux, J.P., (2007). Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility, *LCA Report (EPFL University)*, 2007

Rasmussen, K.B.; Capkun, S. (2007). Implications of radio fingerprinting on the security of sensor networks, *Proceedings of International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pp. 331-340, Sep. 2007

Rubinstein, M.G.; Moraes, I.M., Campista, M.E.M., Costa, L.H.M.K., Duarte O.C.M.B. (2006). A survery on wireless ad hoc networks, In: *Mobile and Wireless Communication Networks (from International Federation for Information Processing (IFIP)*), Guy Pujolle (Ed.), Vol. 211, pp. 1-33, Springer, Boston

Sastry, N.; Shankar, U. & Wagner, D. (2003). Secure verification of location claims, *Proceedings of ACM workshop on Wireless Security*, pp. 1-10, 2003, ACM Press

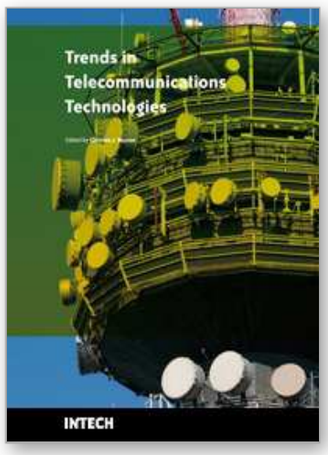
Sayad Haghighi, M. ; Mohamedpour, K. (2008). Securing wireless sensor networks against broadcast attacks, *Proceedings of International Symposium on Telecommunications*, pp. 49-54, Aug. 2008

Shokri, R.; Poturalski, M., Ravot, G., Papadimitratos, P., Hubaux, J.P. (2008). A low-cost secure neighbor verification protocol for wireless sensor networks, *LCA Report (EPFL University)*, Oct. 2008

Singelee, D.; Preneel, B. (2005), Location verification using secure distance bounding protocols, *Proceedings of IEEE International Conference on Mobile Ad hoc and Sensor Systems Conference,* pp. 840-846, Nov. 2005

Zhu, S.; Setia, S. & Jajodia, S. (2003). *LEAP: efficient security mechanims for large-scale distributed sensor networks*, *Proceedings of ACM conference on Computer and Communications Security*, pp. 62-72, May 2003

Zhu, S.; Setia, S. & Jajodia, S. (2006). LEAP+: Efficient security mechanisms for large-scale distributed sensor networks, *ACM Transactions on Sensor Networks,* Vol. 2, No. 4, Nov. 2006, pp. 500-528



Trends in Telecommunications Technologies

Edited by Christos J Bouras

ISBN 978-953-307-072-8
Hard cover, 768 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

The main focus of the book is the advances in telecommunications modeling, policy, and technology. In particular, several chapters of the book deal with low-level network layers and present issues in optical communication technology and optical networks, including the deployment of optical hardware devices and the design of optical network architecture. Wireless networking is also covered, with a focus on WiFi and WiMAX technologies. The book also contains chapters that deal with transport issues, and namely protocols and policies for efficient and guaranteed transmission characteristics while transferring demanding data applications such as video. Finally, the book includes chapters that focus on the delivery of applications through common telecommunication channels such as the earth atmosphere. This book is useful for researchers working in the telecommunications field, in order to read a compact gathering of some of the latest efforts in related areas. It is also useful for educators that wish to get an up-to-date glimpse of telecommunications research and present it in an easily understandable and concise way. It is finally suitable for the engineers and other interested people that would benefit from an overview of ideas, experiments, algorithms and techniques that are presented throughout the book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen