Chapter 1
Formal Model for (k)-Neighborhood Discovery Protocols ⋆

Raphaël Jamet and Pascal Lafourcade

Abstract :
Neighborhood discovery is a critical part of wireless sensor networks, yet little work has been done on formal verification of the protocols in presence of both intruder nodes and mobility. We present a formal trace-based model to verify protocols doing neighborhood discovery, and we provide a formal definition of (1)-neighborhood and (k)-neighborhood. We also analyze a protocol from the literature, and show some conditions needed for its correctness. Finally, we present the groundwork for a protocol which discovers (k)-neighborhood based on (1)-neighborhood data under some assumptions, and prove that it remains secure even if an intruder interferes.

Key words: Formal verification, Wireless Sensors Networks, Communication Protocols, Neighborhood Discovery.

1.1 Introduction

The number of wireless networks is ever increasing. Cellular phones are now more common than wired ones, the number of mobile connections to Wireless Local Area Networks (WLANs) is increasing, and wireless devices are now used everyday at homes, companies and administrations. Wireless devices have become so small and cheap that they can be used in sensor networking applications such as environmental or building monitoring. These devices communicate by relaying packets of other devices across multiple wireless links (hops). Since the devices are often mobile the topology of the network can change over time. Even if the nodes are static, a node can disappear from the network due to battery shortage, temporary interference or physical damage, which will also alter the network configuration. As a result one of the main issues for these networks is that each node must discover or rediscover which nodes are within its communication range: a process called *neighborhood discovery.*

Neighborhood discovery protocols are basic components in mobile wireless systems. Knowledge of the neighbors of a node is for instance essential for routing protocols. The goal of a neighborhood discovery protocol is to identify as neighbors only those devices that are really neighbors, even in presence of intruders.

Université Grenoble 1, CNRS,Verimag, FRANCE

⋆ This work was supported by Projects TERRA and ARESA2

Designing a secure neighborhood discovery protocol is not an easy task, as illustrated in [And01] with the famous *"MIG-in-the-middle"* attack mounted in the late 1980s. In this attack, Angolan MIG airplanes were able to be impersonate a South African unit by relaying challenge messages from South African defense to another South Africa units, which then answered the challenge for them. The Angolan MIGs could therefore bomb their targets without being attacked by the South African air defense. A secure neighborhood discovery system would have prevented such a problem, by detecting that the MIG is not actually in range for the authentication protocol, and then preventing further impersonation by enemy forces. In [PPS+08], the authors survey the question of secure neighborhood discovery, providing definitions of neighborhood types and neighbor discovery protocol properties. They also describe different attacks against wireless networks.

We consider that two nodes are neighbors if they can communicate directly together, without the help of another node. In this, they are closely related to distance bounding protocols, which are designed to determine a upper bound on the distance separating two nodes. Our aim is to define formally this neighborhood property. The ability to verify that two nodes are neighbors can be used to prevent some attacks like wormhole attacks [HPJ03]. In this context, we consider that an attack is any situation in which two nodes communicate together believing that they are neighbors when in fact, other possibly malicious nodes are relaying the communication, whether nodes are MIGs, smart-cards or wireless sensors.

One aim of modern security is to provide rigorous guaranties that the designed system satisfies the required security properties. It is about giving rigorous models for systems, formal definitions of security properties and rigorous proofs under precisely identified assumptions. Proving that a protocol is secure is not an easy task, even for simple and short protocols. In 1996 G.Lowe [Low96] found an attack on the famous Needham-Schroeder protocol [NS78] seventeen years after his publication. He found this flaw using his automatic verification tool Casper [Low97] based on the CSP (Communicating Sequential Processes) model checker FDR [Hoa85, Sch96, RSG+00]. The protocol proposed by Needham and Schroeder was proven secure by using a formalism called BAN logic under different assumptions on the intruder model: only for one single execution of the protocol. The flaw proposed uses two parallel executions of the same protocol with different participants and assumes the so-called *perfect encryption hypothesis*. More precisely it means that the only way to uncrypt a cipher text is to know the inverse key. This hypothesis abstracts the cryptography in order to detect "logical flaw" due to all possible interleavings of different execution of the protocol. In this formalism:

- messages are represented by terms build over a signature
- intruder controls the network
- perfect encryption assumption is done
- intruder has a limited abilities.

The intruder capabilities are usually represented by the so-called *Dolev-Yao* intruder model [DY81]. This intruder model captures the perfect encryption hypothesis. This approach is called *symbolic* by contrast to the *computational* approach proposed by the cryptographer. The discover of the "logical" attack by G. Lowe shows that even experts can miss some flaws even on small protocol (only three message exchanges) and certainly underestimate the complexity of the security analysis of such protocols. It also indicates that automatic analysis is critical for assessing the security of cryptographic protocols, because humans can not reasonably verify their correctness. Hence symbolic and automatic verification of cryptographic protocols became a main and active topic in security. In [BCM11] one can find a survey of formal approaches for proving security protocols.

Motivational scenario

Let us consider a wireless network, intended to assist firefighters during an operation inside disaster sites. This kind of WSNs are being investigated since a few years. For instance in [WBRW07], Wilson et al. present the SmokeNet WSN, used to provide firefighters critical data about the building currently on fire, and also monitor personnel health and position. SmokeNet itself is a pre-existing network consisting of small sensors (motes) scattered through the building. All the motes regularly emit localization data. Additionally, some motes also monitor smoke, temperature, carbon monoxide levels, which combined give the possibility to follow the progress of an hypothetical fire. The applications are designed to resist node failures, and thus empty batteries, physical damage, and other causes of disruption are not critical. Firefighters usually are organized in a command post, outside of the building, and agents inside the building. The command post has access to the SmokeNet data, and can thus monitor the progress of the fire. Agents inside do not have the time to sort through all of this data, and instead must be relayed orders from the command post. This relaying happens through a network which is built with the cooperation of both the pre-existing motes and the equipment worn by each firefighter. They also consider the possibility of relay motes dropped by firefighters as they progress into the building.

Taking all of this into account, we propose a few changes in the SmokeNet system and firefighters equipment which would enable new features. In certain kinds of incidents (chemical, for instance), firefighters are required to stay within a certain range of each other (rescue distance). We can use SmokeNet for this, but the current localization systems in place in the network requires pre-existing calibration of the motes, maps, and mote placement information. This may not be available in all buildings, and to work around this, our proposition does not need this data.

Let us assume that the command center wants firefighters to stay at 40 meters of each other, but sensors can only communicate with another sensor 20 meters away at most. How could we guarantee that if the distance between two firefighters gets higher than 40 meters, then the sensors detect it and alert their wearers ? By ensuring that two worn nodes can always communicate directly, or through one other node. Alerts when the distance is less than 40 meters are not much of an issue, but if the distance is dangerous, the sensors must react. This is where neighborhood detection and distance bounding gets useful. If we can use a proven protocol which detects the distance between nodes, we can keep track of the firefighter rescue radius. If we know the communication range, using the mere possibility of communication is similar. All of this does not require any localization data. And since we do not want to be restricted by the range of the sensors, it should be possible to use more than two nodes in the distance detection.

To worsen things, we would like to be able to guarantee that within some known limits, these protocols still provably work in presence of serious malfunctions. In order to model this, we choose a worst-case assumption, malicious sensors which actively try to disrupt the protocol. This is a more demanding requirement than simple faults in sensors.

Contributions:

One of our main goals is to formally define the concept of neighborhood. In order to achieve it and based on the observation that the neighborhood is a physical

property[2], we distinguish two layers: the physical layer representing physical characteristics of the communications (radio for instance) and the abstract layer modelling node behavior. Then we formalize the communication between nodes by a trace-based model inspired by the symbolic approach proposed by Paulson in [Pau98]. For this, we introduce *send* and *receive* events for the two layers for modelling the communications between nodes. We propose a system of rules for modeling exchanges between nodes given a topology which represents the environment and position between nodes. Moreover, we notice that most of the neighborhood discovery protocols use time measurements and can work even for mobile nodes, so we add timestamps in all our events. We also explain how it is possible to model a protocol in our framework, by generating new rules according to the specification of a given protocol. Finally we give a deduction system which models the abilities of an intruder. Using all these ingredients we can define the property of neighborhood by generating a special event called *END* when a nodes conclude that he is neighbor of another node. Consequently, an attack is the situation where a nodes concludes that another node is his neighbor according to the protocol but indeed they are not able to communicate directly. A protocol is secure for this property if for any execution in presence of intruder, in a given topology, there exists no attack. Then we extend it to (k)-neighborhood, *i.e.* we propose a formal definition for modelling that two nodes can communicate with k hops. We give examples illustrating how our framework works to discover attack on a protocol given an intruder and a topology or to prove the security of a protocol under some assumptions. Finally we propose a protocol, called *Sharek* protocol, for discovering the (k)-neighborhood of a node based on a secure (1)-neighborhood secure protocol. This protocol can be used to securely discover the set of neighbors of neighbors, and further. It can help to determine the dominating sets of a wireless network, as explained in [DB97, WL99, BMSU01]. The main goal of this protocol is to present, as far as we know, the first secure protocol which can be used to securely discover the (k)-neighborhood of a node. Of course using our formalism, we then provide a formal proof of security for this protocol.

Related works:

Neighborhood discovery and analysis of protocols are active research topics in wireless networks security. One of the first neighborhood discovery protocol was given by S. Brands and D. Chaum in [BC94]. Later other works have approached the problem differently using for instance directional antennas [VKT05, DKPR11], probabilistic protocols and challenge-response in RFID context [HK05], or specific protections against some attacks by analyzing network topology based on time of flight of messages in [SPR+09].

One of the first formal verification of neighbor discovery protocol is the work done by Meadows et al. in [MPP+06]. In this paper they developed a formal methodology to prove properties of distance bounding protocols [MPP+06]. They extend the authentication logic to reason about distance bounding property and they extend the protocol of S. Brands and D. Chaum. However, they do not consider (k)-neighborhoods, and deal only with static nodes.

In [PPH08a], the authors investigate the possibility of neighborhood detection. They consider several transmission speeds, directional emissions, localization and clock synchronization, and conclude by proving that time-based protocols cannot securely detect neighborhoods if and only if intruders can forward faster than legitimate nodes. They also consider protocols which use location data.

[2] We remark that is not the case for the authentication which is a property based on exchanged messages.

Another formal approach for distance bounding protocols is introduced by Basin et al. in [SSBC09]. Their approach also uses notions of distances, time and events-trace. They use the Isabelle/HOL proof assistant [NPW02] to check results from their model. They apply their model on three security protocols, the authentication ranging protocol [CH05], the distance bounding protocol of [BC94], which calculates distance between two nodes using communication time and finally the TESLA protocol of [PCTS02]. In their work they only consider static nodes and do not use (k)-neighborhoods.

In [TSG10] the authors present a systematic technique for verifying that location discovery protocols satisfy locale authentication whereby an entity can authenticate the physical location of a device, even in the presence of malicious adversaries. They extend the strand space theory with a metric that captures the geometric properties of time and space. They prove that several prominent location discovery protocols including GPS do not satisfy the local authentication goal and analyze a location discovery protocol that does satisfy the goal under some reasonable assumptions.

In some recent works, C. Cremers et al [CRC11] propose distance bounding protocols resistant to distance hijacking attacks. The authors in [PPH08b] and [PPH07] build a formal model based on traces. They apply their model on a protocol they introduced and on the temporal packet leash protocol [HPJ03]. However their method does not take the mobility of the nodes into account.

More recently, in [ABK$^+$11] the authors propose a framework for analyzing distance bounding protocols in RFID, using a computational approach and proposed a white-box and black-box point of view. Our work differs from those by using a symbolic approach which abstracts a way most of the cryptographic considerations.

Outline:

In Section 1.2, we present our model. We first give the network representation, the events and traces definitions, and show how to specify a protocol and an intruder. We also illustrate by a running toy example all our definitions. Then in Section 1.3, we define the neighborhood property. We extend the usual notion of (1)-neighborhood to (k)-neighborhood. In Section 1.4, we illustrate our approach by analyzing a protocol proposed in [PPH08a]. We prove that it is secure against a certain class of intruders, but is vulnerable against stronger intruders. In Section 1.5, we propose a secure protocol to build the (k)-neighborhoods of a node, based on any secure (1)-neighborhood protocol. Finally we prove it is secure against a class of intruders, before concluding in the last section.

1.2 Timed Model

Let us consider the topology depicted in Figure 1.1. In this case, if communication times are proportional to the distances, then, due to signal reflection on the wall and to the presence of a wall between A and B, we have that the transfer time between A and I plus the transfer time between I and B is strictly smaller than the transfer time between A and B. This simple situation shows that in some setting triangular inequality may not always hold in wireless connections. It is why we consider transfer time and not distances in our approach.

Moreover our model is only relying on the possibility of communication between two nodes. We consider the time needed to transmit a message between two nodes. By consequence two nodes are neighbors if they can communicate directly, *i.e.* their transmission time is finite. Moreover, in order to analyze the security of the

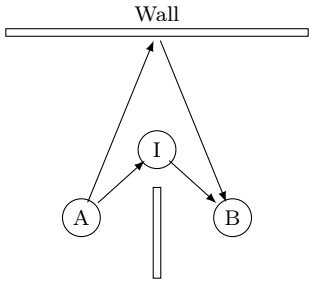

Fig. 1.1 Triangle inequality counter-example.

protocols we consider either honest nodes, which follow the protocol, or malicious nodes. We illustrate our model by using a flawed toy example: Ping-Pong protocol. This protocol finds neighbors of a node by simply sending them a solicitation (the Ping), to which neighbors will answer with their name and a Pong, denoted by the tuple $\langle B, Pong \rangle$. Once the Pong is received, the protocol claims that the initiator and the node whose identity is in the Pong message are neighbors. It is obviously not secure, but will help us to present our framework. A high-level description of this protocol is given in Figure 1.2.

$$A \xrightarrow{Ping} B$$

$$A \xleftarrow{\langle B, Pong \rangle} B$$

Fig. 1.2 Ping-Pong protocol communications diagram

In this section we start by formally defining characteristics of a network and nodes. Then we model behaviors of nodes using two family of events with timestamps. We explain using rules describing a given protocol how to build traces, which are sequences of events. We model by a set of rules the intruder capabilities. Finally, we explain two realistic assumptions, which allow us to consider mobility in our approach.

1.2.1 Networks and Nodes

Wireless Sensors Networks are composed of several nodes, that are small devices usually equipped with sensors, a battery and a radio. They use their radio for sending their measurements through the network. To keep each device cheap, they have a small memory and limited computing power. Each node has an identity and can also possess pre-shared cryptographic keys depending on the application. A network is composed of a set of nodes (identified by an unique number) and a topology, which represents communications between nodes.

Definition 1 (Network). A network $\mathbf{N}$ is defined by two components $(V, \mathbf{T})$ where:

- V denotes the set of all node identities, which is partitioned into two sets: $V_{\mathcal{P}}$ denotes all the honest nodes who are following the protocol and $V_{\mathcal{I}}$ represents the intruder nodes which are malicious.

- $\mathbf{T}$ represents the matrix containing the time of communication between two nodes. More precisely, $\mathbf{T}_{A,B}$ denotes the time that a message takes starting from node A to reach node B. If A can not reach directly B then $\mathbf{T}_{A,B}$ is equal to $+\infty$. We also require that the communication time between a node and itself is null: $\forall A \in V, \mathbf{T}_{A,A} = 0$.

We denote honest node identities by $A, B, C, ...,$ and $I, J, ..$ for intruders. By extension, we will often refer to nodes by their identity (for example, node A).

This definition models only static networks. In order to express the possible movements of nodes, we need to take into account the changes in topology over time, and so we extend the previous definition by making the communication time depend on time. $\mathbf{N} = (V, \mathbf{T})$ becomes $\mathbf{N} = (V, \mathbf{T}(t))$, where the element of the matrix are functions with parameter t. Each element $\mathbf{T}_{A,B}(t)$ models how much time a message **emitted** by the node A at time t takes to reach node B. If needed, we use $\mathbf{T}_{A,B}(t)$ instead of $\mathbf{T}_{A,B}$.

We denote $\mathcal{N}$ the set of all possible networks, which takes into account any number of nodes, any number of intruders, and any possible evolution of the connections over time. Here are a few examples of useful network families:

- $\mathcal{N}_0$ is the family of networks where no intruder is present at all. This allows us to show that a protocol is insecure even when no intruder is present.

$$\mathcal{N}_0 = \{\mathbf{N} = (V, \mathbf{T}(t)) \in \mathcal{N} \mid V_{\mathcal{I}} = \emptyset\}.$$

- $\mathcal{N}_{sym}$ is the family of networks where message transfer times are symmetrical.

$$\mathcal{N}_{sym} = \{\mathbf{N} = (V, \mathbf{T}(t)) \in \mathcal{N} \mid \forall t, \forall A, B \in V, \mathbf{T}_{A,B}(t) = \mathbf{T}_{B,A}(t)\}.$$

- $\mathcal{N}_{fully\_connected}$ is the family of networks where each node can communicate with any other node at any time.

$$\mathcal{N}_{fully\_connected} = \{\mathbf{N} = (V, \mathbf{T}(t)) \in \mathcal{N} \mid \forall t, \forall A, B \in V, \mathbf{T}_{A,B}(t) \neq +\infty\}.$$

- $\mathcal{N}_{still}$ is the family of networks where there is no change in transfer times depending on time, or in other words a static network.

$$\mathcal{N}_{still} = \{\mathbf{N} = (V, \mathbf{T}(t)) \in \mathcal{N} \mid \forall t_1, t_2, \forall A, B \in V, \mathbf{T}_{A,B}(t_1) = \mathbf{T}_{A,B}(t_2)\}.$$

1.2.2 Events

Neighborhood is a physical property, which is linked to communication channels. In order to isolate physical communications and abstract commands, we consider two distinct layers, one of which is strictly restricted to model physical behavior, and the other corresponding to the abstract behavior of the nodes, which correspond to computations performed by a node. In order to model communication between nodes and behaviors of a node on these two layers we define the following events:

- $send_\phi(A, m, t)$: A transmits m at time t using the physical layer.
- $recv_\phi(A, m, t)$: A receives m at time t using the physical layer.
- $send_\alpha(A, m, t)$: A orders the transmission of m at time t using the logical layer.
- $recv_\alpha(A, m, t)$: A received and processed m at time t using the logical layer.

Physical events are annotated by ϕ and abstract events are annotated by α. In order to construct meaningful sequences of events, nodes need to be able to know when a message was sent or received at the physical layer. It is why each event has

a timestamp. In the Ping-Pong protocol the first action performed by A is modeled by the event $send_\alpha(A, Ping, t_0)$ and $send_\phi(A, Ping, t_1)$, where t_0 is the time when the node A transmit to the radio the message $Ping$ and t_1 is the time when the radio of the node A emits the message.

We also introduce an extra event $END(N_1, ..., N_k, t_{prop}, t)$, which models the fact that a protocol ended well at time t, and concludes something about the nodes $N_1, ..., N_k$ at time t_{prop}. For example, if we keep our objective of neighborhood discovery protocols, that conclusion could be the possibility of direct communication between the specified nodes at the time of the property. In our running example, A would like to conclude after the exchange of messages that he has B as neighbor. We model it by the event $END(A, B, t_2, t_4)$, where t_4 is the time when the protocol ended and t_2 is the time when the node A concludes that he has B as neighbor. We decide to store in the event $END(A, B, t_{prop}, t)$ both the time t_{prop} when the message was sent by A to B, because if B receives it they are neighbors and the time t when A receives a reply from B, because here also they are neighbors.

1.2.3 Rules and Traces

We now explain how to construct possible communications between different nodes. We build traces which are sequence of events. In order to generate these sequence we use three sets of rules which model the protocol, the communications between nodes and also the intruder. We first present how to use a set of rules for building a trace. We illustrate how to model a protocol with our running example, before showing set of communication rules which are always present in our model. Finally we give the description of intruder rules in Section 1.2.4.

1.2.3.1 Building traces from rules

Our approach is based on the trace-based modeling presented originally by Paulson for cryptographic protocols in [Pau98]. The rules represents a step in the construction of a trace and has the following form:

$$(R)\frac{H_1 \dots H_n}{C}$$

where R is the name of the rule, $H_1 \dots, H_n$ are the hypothesis that have to be satisfied in order to produce the conclusion C.

A trace is a set of events which models the communication between nodes during the execution of a protocol in a given network and a given intruder behavior. Each node can emit and receive messages on the abstract layer, messages can be transferred from a layer to another, and lastly the communication between two nodes's physical layers allow messages to go from a node to another. Those three processes are modeled with rules which are common to all networks (details are given in the next paragraphs). We remark that the traces are sets of events, not sequences as in Paulson's model: since we have timestamps for each event, the order is implicit.

Intruders are modeled by $\mathcal{I}$, a set of rules that describes their abilities and behaviors. A precise description of the intruder model is given in 1.2.4. The behavior of a node which respects a protocol is modeled by a set of rules denoted $\mathcal{P}$. Usually protocols are specified at the logical layer, and generate an *END* event when the protocol finishes to model what the protocol claims. We denote by $S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ the set

of all traces, *i.e.* possible executions, built by successive applications of rules of the system.

We now write the three rules of $\mathcal{P}_{PingPong}$ needed to model this protocol.

$$(PingPong\_1)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}{(tr \cup \{send_\alpha(A, Ping, t)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(PingPong\_2)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad recv_\alpha(A, Ping, t_1) \in tr \qquad t_1 \leq t_2}{(tr \cup \{send_\alpha(A, \langle A, Pong\rangle, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(PingPong\_3)\frac{\begin{array}{c} tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \\ send_\alpha(A, Ping, t_1) \in tr \qquad send_\phi(A, Ping, t_2) \in tr \\ recv_\alpha(A, \langle B, Pong\rangle, t_3) \in tr \qquad t_1 \leq t_2 \leq t_3 \leq t_4 \end{array}}{(tr \cup \{END(A, B, t_2, t_4)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

The two first rules correspond to the two rules of the protocol, properly specified with timestamps on the abstract layer. The last rule raises the *END* flag, which models the fact that the protocol reached a conclusion about A and B.

1.2.3.2 Communication Rules

In Figure 1.3, we present the set of rules inherent to all networks.

$$(Begin)\frac{}{\emptyset \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(Con0)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad send_\alpha(A, m, t_1) \in tr \qquad t_1 + \delta/2 \leq t_2}{(tr \cup \{send_\phi(A, m, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(Con1)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad recv_\phi(A, m, t_1) \in tr \qquad t_1 + \delta/2 \leq t_2}{(tr \cup \{recv_\alpha(A, m, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(Phy)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad send_\phi(A, m, t_1) \in tr \qquad \mathbf{T}_{A,B}(t_1) \leq (t_2 - t_1)}{(tr \cup \{recv_\phi(B, m, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

Fig. 1.3 Basic intruder knowledge building rules

The first rule is the initialization of a trace which is made by axiom $(Begin)$. It means that all trace starts empty. Application of other rules adds events in order to obtain a complete execution trace. To generate events from one layer to the other, there are two rules $(Con0)$ and $(Con1)$, where δ denotes the delay a node needs to relay a message. Finally we have the rule (Phy) which allows messages to go from a node's physical layer to another node, in a time greater than or equal to $\mathbf{T}_{A,B}(t_1)$. This rule may be applied multiple times for a single send event, to model signal reflections which may cause a message to be received twice or more, as shown in Figure 1.4. It is why there is no upper bounds on t_2. We also assume that all the nodes have access to a $New\_Nonce()$ function, which returns a fresh random value at each new call.

We consider our running example and give the trace corresponding to the following scenario. A ordered the broadcast of message "$Ping$", which was transferred to the physical layer. Then B receives the message in their physical layer, which later on got shifted onto the logical layer. Finally B replies to A, his name and "$Pong$", and A concludes that he is neighbor with B. We obtain the following trace assuming

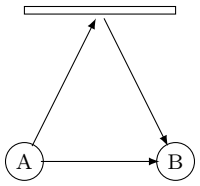

Fig. 1.4 Network where the same message hits the same node twice.

that $\delta = 1$ and the communication between A and B is symmetric and for all t, $\mathbf{T}_{A,B}(t) = 10$. For the *END* event, see rule $(PingPong_3)$ for the choice of $t_{prop} = 1$.

- $send_\alpha(A, Ping, 0)$
- $send_\phi(A, Ping, 1)$
- $recv_\phi(B, Ping, 11)$
- $recv_\alpha(B, Ping, 12)$
- $send_\alpha(B, \langle B, Pong\rangle, 12)$
- $send_\phi(B, \langle B, Pong\rangle, 13)$
- $recv_\phi(A, \langle B, Pong\rangle, 23)$
- $recv_\alpha(A, \langle B, Pong\rangle, 24)$
- $END(A, B, 1, 24)$

The first time stored in the event $END(A, B, 1, 24)$ corresponds to the time when A sent a message to B and the second one is the time when A receives an answer from B. With these times we catch all the times where the nodes are neighbors. It helps us to cover for instance the following situation: node B receives a message from A qnd moves out to the range of A during a while, then he answers when he is again at communication distance of A.

1.2.4 Intruder Rules

Intruder capabilities are described by the set of rules denoted $\mathcal{I}$, common to all the intruder nodes. $\mathcal{I}$ describes both what an intruder node can know, and what it can do.

1.2.4.1 Intruder knowledge

Each of the intruder nodes has some knowledge, which is built with rules, alongside the execution of the protocol. That knowledge depends on both what the intruder node was able to hear, and what it knew before the protocol executes. This knowledge is specific to a given node: what a node learns does not instantly propagates to other intruders' memories.

We call that knowledge IK_I for the node I (and IK_J for J, and so on). More precisely, $IK_I(tr, t)$ represents what I knows at time t assuming the events from trace tr happened, and $\widehat{IK}_I(tr, t)$ represents what I can deduce from $IK_I(tr, t)$. We denote $IK_I(\emptyset, 0)$ the initial knowledge of the intruder I. For instance, in a classical outsider attack, all of the $IK_I(\emptyset, 0)$ would contain the identifiers of the nodes in the network and some fictive ones.

On the abstract level, we adapt the classical Dolev-Yao intruder models [DY83] for building the intruder knowledge. The adapted Dolev-Yao deduction system is given by the rules in Figure 1.5 (containing the basic knowledge-building rules) and

$$(IK\_init)\frac{m \in IK_I(\emptyset, 0)}{m \in IK_I(tr, t)}$$

$$(IK\_hat\_promotion)\frac{m \in IK_I(tr, t)}{m \in \widehat{IK_I(tr, t)}}$$

$$(IK\_left)\frac{\langle m, n\rangle \in \widehat{IK_I(tr, t)}}{m \in \widehat{IK_I(tr, t)}}$$

$$(IK\_right)\frac{\langle m, n\rangle \in \widehat{IK_I(tr, t)}}{n \in \widehat{IK_I(tr, t)}}$$

$$(IK\_pair)\frac{m \in \widehat{IK_I(tr, t)} \qquad n \in \widehat{IK_I(tr, t)}}{\langle m, n\rangle \in \widehat{IK_I(tr, t)}}$$

Fig. 1.5 Basic intruder knowledge building rules

Figure 1.6, which describes the rules for symmetric and asymmetric cryptography, and nonce generation. To avoid confusion, all symmetric keys are labelled k, and asymmetric keys are couples (sk, pk) and encryptions are represented by $\{m\}_k$. Notice that an intruder I can increase his knowledge using a message m at time t_2 only if there is a $recv_\alpha(I, m, t_1) \in tr$ with $t_1 \leq t_2$.

$$(IK\_decrypt\_sym)\frac{\{m\}_k \in \widehat{IK_I(tr, t)} \qquad k \in \widehat{IK_I(tr, t)}}{m \in \widehat{IK_I(tr, t)}}$$

$$(IK\_encrypt\_sym)\frac{m \in \widehat{IK_I(tr, t)} \qquad k \in \widehat{IK_I(tr, t)}}{\{m\}_k \in \widehat{IK_I(tr, t)}}$$

$$(IK\_decrypt\_asym\_sk)\frac{\{m\}_{sk} \in \widehat{IK_I(tr, t)} \qquad pk \in \widehat{IK_I(tr, t)}}{m \in \widehat{IK_I(tr, t)}}$$

$$(IK\_encrypt\_asym\_sk)\frac{m \in \widehat{IK_I(tr, t)} \qquad sk \in \widehat{IK_I(tr, t)}}{\{m\}_{sk} \in \widehat{IK_I(tr, t)}}$$

$$(IK\_encrypt\_asym\_pk)\frac{m \in \widehat{IK_I(tr, t)} \qquad pk \in \widehat{IK_I(tr, t)}}{\{m\}_{pk} \in \widehat{IK_I(tr, t)}}$$

$$(IK\_nonce)\frac{N_I = New\_Nonce()}{N_I \in \widehat{IK_I(tr, t)}}$$

Fig. 1.6 Cryptography-related intruder knowledge building rules

1.2.4.2 Intruder Actions

The rules to link the transmission model and the intruder knowledge are given in Figure 1.7. (IK_recv) is used so that intruders can hear communications, and learn from them. The rule $(Intrude\_replay)$ models how the intruder can replay messages it heard. The rule $(Intrude\_forge)$ represents how an intruder can deduce and build new messages from his knowledge and send them. The distinction between those two rules allows simple intruders which would only be able to relay messages, formally described by not including $(Intrude\_forge)$ in $\mathcal{I}$. Usually, $\mathcal{I}$ can forge the same messages as the protocol $\mathcal{P}$, in the case of an intruder with a large enough initial knowledge $IK_I(\emptyset, 0)$.

$$(IK\_recv)\frac{recv_\alpha(I, m, t_1) \in tr \qquad t_1 \leq t_2}{m \in IK_I(tr, t_2)}$$

$$(Intrude\_replay)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad m \in IK_I(tr, t_1) \qquad t_1 \leq t_2}{(tr \cup \{send_\alpha(I, m, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(Intrude\_forge)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad m \in \widehat{IK}_I(tr, t_1) \qquad t_1 \leq t_2}{(tr \cup \{send_\alpha(I, m, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

Fig. 1.7 Intruder knowledge input and usage rules

1.2.5 Mobility

Mobility of nodes is already included in the model, since the values of $\mathbf{T}(t)$ varies with time. However, we need to add two realistic assumptions in our model.

Assumption 1 *A node moves much slower than a message.*

This allows us to assume that values in $\mathbf{T}(t)$ takes into account the (negligible) movements of the nodes during the subsequent transfer time (since $\mathbf{T}(t)$ stores a fixed configuration of nodes and their movement depending on time). This is straightforward: radio messages travel near the speed of light, while wireless sensors are usually slower.

Assumption 2 *δ is in the same order of magnitude of value as the message transfer time.*

This assumption is related to rules $(Con0)$ and $(Con1)$. Remember that δ is the time needed to forward a message.

Taking into account the previous assumption and this one, we can deduce that sending and relaying messages does not take a significant time with regard to node movement. To restate these two rules in a less formal way, we assume that it is possible that during message transfers and relayings, the values of $t_{transfer}(A, B, t)$ do not change significatively, no matter which $\mathbf{N}$ is used.

On the other hand, it is always possible to add arbitrary delays to operations: see rules (Phy), $(Con0)$, $(Con1)$ and $(Intrude\_replay)$. For instance, these delays may be used in an attack by allowing synchronization of multiple sessions of a protocol.

1.3 Neighborhood and (k)-Neighborhoods

1.3.1 Neighborhood

A node A is *neighbor or* (1)*-neighbor to the node* B *at time* t if $\mathbf{T}_{A,B}(t)$ is finite which means that A is in communication range of B.

Definition 2 (Neighborhood). Let $\mathbf{N} = (V, \mathbf{T}(t))$ be a network, the *neighborhood of a node at time* t, denoted by $\mathbf{Ng}^1_A(t)$, is the set of nodes that A can reach with a message sent at time t. It is formally defined by:

$$\mathbf{Ng}^1_A(t) = \{X \mid X \in V \wedge \mathbf{T}_{A,X}(t) < +\infty\}.$$

If $tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ is a trace, we denote by $\widehat{tr}$ the set of all possible traces that can be inferred from tr using rules defined previously. It allows us to define $t_{transfer}(A, B,)$, the smallest time possible needed to send a fresh message m from A to B at time t using only communication rules.

We also define a protocol $\mathcal{P}_{Forward}$ and an intruder $\mathcal{I}_{Forward}$ which consist in the following unique rule which makes nodes forward message on abstract layer:

$$(Forward)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad recv_\alpha(A, m, t_1) \in tr \qquad t_1 \leq t_2}{(tr \cup \{send_\alpha(A, m, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

Definition 3 ($t_{transfer}(A, B, t)$). Let $\mathbf{N} = (V, \mathbf{T}(t))$ be a network, and $tr = \{send_\phi(A, m, t)\}$ be a trace in $S_{\mathbf{N},\mathcal{I}_{Forward},\mathcal{P}_{Forward}}$. We define $t_{transfer}(A, B, t)$ by:

- $t_{transfer}(A, B, t) = \min\{x - t \mid recv_\phi(B, m, x) \in \widehat{tr} \subseteq S_{\mathbf{N},\mathcal{I}_{Forward},\mathcal{P}_{Forward}}\}$.
- If the event $recv_\phi(B, m, t)$ does not belong to $S_{\mathbf{N},\mathcal{I}_{Forward},\mathcal{P}_{Forward}}$ then we state that $t_{transfer}(A, B, t) = +\infty$.

We use $\mathcal{I}_{Forward}$ and $\mathcal{P}_{Forward}$ to consider the simplest situation, where node can only forward messages. The send event introduced in all generated trace allows us to build all possible routes taken by this message. Note that the $(Forward)$ rule does not impose to forward the message immediately.

$t_{transfer}(A, B, t)$ is different than $\mathbf{T}_{A,B}(t)$, because it can take into account multiple hops, and the relay times needed in the intervals. Also, there may be routes which are faster than the direct ones (with some conditions on δ and $\mathbf{N}$, as in the first example in Figure 1.1).

We denote by $t_{max\_emitter}$ the maximum positive finite time in $\mathbf{T}(t)$, *i.e.* the maximal communication time of two connected nodes. If we assume that $\delta > t_{max\_emitter}$ then we are able to characterize the definition of $\mathbf{Ng}^1_A(t)$ by relations between $t_{max\_emitter}$ and $t_{transfer}(A, B, t)$.

Lemma 1. *Let* $\mathbf{N} \in \mathcal{N}$ *be a network containing at least the two nodes* A *and* B*. If we assume that* $\delta > t_{max\_emitter}$*, the following properties are equivalent:*

1. $B \in \mathbf{Ng}^1_A(t)$
2. $t_{transfer}(A, B, t) \leq t_{max\_emitter}$

Proof. We prove a double implication.

- $1 \Rightarrow 2$: Let us assume that $B \in \mathbf{Ng}^1_A(t)$. We know there exists a sequence of events in $S_{\mathbf{N},\mathcal{I}_{Forward},\mathcal{P}_{Forward}}$ containing only the event $send_\phi(A, m, t)$. Since $B \in \mathbf{Ng}^1_A(t)$, we can generate a valid trace $tr' = \{send_\phi(A, m, t), recv_\phi(B, m, t+\mathbf{T}_{A,B}(t))\}$, using (Phy). Therefore, $t_{transfer}(A, B, t) \leq \mathbf{T}_{A,B}(t)$. We now prove that this time is the smallest value for $t_{transfer}(A, B, t)$. We assume the opposite $t_{transfer}(A, B, t) <$

$\mathbf{T}_{A,B}(t)$ meaning that $t' - t < \mathbf{T}_{A,B}(t)$ where $recv_\phi(B, m, t')$. The only other way to generate a trace containing $recv_\phi(B, m, t')$ would be to consider that there exists a node X who has sent m to B (using rule (Phy) again). In other words there is a trace containing $send_\phi(X, m, t_{s_X})$ with $t' > t_{s_X}$. Since m is fresh, this event can only be generated from a trace containing a $recv_\phi(X, m, t_{r_X})$ event (using rules $(Con0)$ and $(Con1)$), and we know that $t_{s_X} - t_{r_X} \geq \delta > t_{max\_emitter}$ (by hypothesis). Moreover, also due to the freshness of m, we have $t_{r_X} > t$. We deduce that $t' - t > t_{s_X} - t_{r_X} > t_{max\_emitter}$. This leads to a contradiction with $t' - t < \mathbf{T}_{A,B}(t)$. We conclude that $t_{transfer}(A, B, t) = \mathbf{T}_{A,B}(t)$ then by definition of $t_{max\_emitter}$ we obtain that $t_{transfer}(A, B, t) \leq t_{max\_emitter}$.

- $2 \Rightarrow 1$: We make a proof by contradiction. We assume that $B \notin \mathbf{Ng}^1_A(t)$, our aim is now to prove that $t_{transfer}(A, B, t) > t_{max\_emitter}$. We distinguish two cases:
  - $t_{transfer}(A, B, t) = +\infty$. In this case, by definition of $t_{max\_emitter}$, we immediately conclude.
  - $t_{transfer}(A, B, t)$ is finite. It means that there exists a way to send a fresh message m between A and B, using a forwarder node which take at least δ units of time. Since $\delta > t_{max\_emitter}$, we conclude that $t_{transfer}(A, B, t) > t_{max\_emitter}$.

  □

As we have already explained, the description of a protocol makes claims (by the mean of an *END* event). In the modeling of the protocol this rule can only be used if the protocol claims that a node is a (1)-neighbor of another node. Our aim is to give a definition which is satisfied only if the protocol is secure: meaning that the protocol reach the flag *END* and the two considered nodes are really neighbor. Our idea for (1)-neighborhood is to prove that if a protocol claims to satisfy (1)-neighborhood then there exists a direct way of communication between the nodes. It is captured by the following definition.

Definition 4. Let $\mathcal{N}$ be a family of networks and $\mathcal{I}$ an intruder. A protocol $\mathcal{P}$ verifies the (1)-neighborhood relation in presence of an intruder $\mathcal{I}$ over $\mathcal{N}$ if and only if $\forall \mathbf{N} = (V, \mathbf{T}(t)) \in \mathcal{N}, \forall A, B \in V$, $\nexists tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ such that $END(A, B, t, t_x) \in tr \wedge B \notin \mathbf{Ng}^1_A(t)$.

To state it otherwise, a protocol does not verify a neighborhood property in presence of an intruder $\mathcal{I}$ over a set of networks $\mathcal{N}$ if and only if there is at least a trace tr in the networks in $\mathcal{N}$, built by the given protocol and intruder rules, such that $END(A, B, t_{prop}, t) \in tr$ and there is no direct communication possibility between those nodes at time t_{prop}. This trace contains an example attack.

Let us illustrate the neighborhoods, the intruder's rules and the formal definition of verifying a property by building an attack on our running example. We define $\mathbf{N}$ by $\forall t, \mathbf{Ng}^1_A(t) = \mathbf{Ng}^1_I(t) = \{A, I\}$ which is simply two nodes in range of each other, one being honest (A) and one malicious (I). We assume for simplicity that nodes do not move and can communicate in finite time.

Regarding the intruder, we consider a basic $\mathcal{I}$ who can forward and forge messages as previously described. We also choose $IK_I(\emptyset, 0) = \{A, I, Z, Ping, Pong\}$, the initial knowledge of I at time 0, with Z an nonexistent node identity.

The scenario of the attack is the following: A starts the protocol by broadcasting $Ping$ using rules $(Begin)$ and $(PingPong\_1)$. The intruder I receives the $Ping$ using (Con_0), (Phy) and (Con_1). Now, I forges $\langle Z, Pong\rangle$, which is known since both parts are in IK_0, and sends that message to A (rule $(Intrude\_forge)$). Then using $(PingPong\_1)$ it is possible to infer $END(A, Z, t_2, t_9) \in tr$ with the appropriate time values. The final trace we obtain corresponds to a possible attack, and contains the following set of events:

$\{send_\alpha(A, Ping, t_1)$, $send_\phi(A, Ping, t_2)$, $recv_\phi(I, Ping, t_3)$, $recv_\alpha(I, Ping, t_4)$,

$send_\alpha(I, \langle Z, Pong\rangle, t_5)$, $send_\phi(I, \langle Z, Pong\rangle, t_6)$, $recv_\phi(A, \langle Z, Pong\rangle, t_7)$, $recv_\alpha(A, \langle Z, Pong\rangle, t_8)$, $END(A, Z, t_2, t_9)\}$, where $t_1 < t_2 < t_3 < t_4 < t_5 < t_6 < t_7 < t_8 < t_9$.

Now let us go back to Definition 4. Here, we have a $\mathbf{N}$, where $Z \notin \mathbf{Ng}_A^1(t_2)$, because there is no Z in the network. We just built a possible trace in this setting including $END(A, Z, t_2, t_9)$, which means the protocol has detected a neighborhood relation between A and Z at time t_2. Therefore the protocol does not verify the neighborhood property with respect to the previously defined intruder $\mathcal{I}$, with his knowledge $IK_I(\emptyset, 0)$, on all the families of networks $\mathcal{N}$ which contain this example $\mathbf{N}$.

.

1.3.2 (k)-neighborhood

We extend our definition in order to determine if a node is neighbor to another one after k hops. For simplicity's sake, we consider that all nodes have the same relaying time δ. We can easily generalize our results with a different time for each node in the network.

Definition 5 ((k)-or-less-neighborhood). Let $\mathbf{N} = (V, \mathbf{T}(t))$ be a network, $\mathbf{Ng}_A^{\leq k}$ (the (k)-or-less-neighborhood of A) is the set of all the nodes B for which there is a path starting at A at time t and ending at B with k hops or less, taking into account the minimal flight time of messages and the relaying time δ. We define it recursively by:

$$\mathbf{Ng}_A^{\leq k}(t) = \left\{B \mid \left(X \in \mathbf{Ng}_A^1(t)\right) \wedge \left(B \in \mathbf{Ng}_X^{\leq (k-1)}\left(t + \mathbf{T}_{A,X}(t) + \delta\right)\right)\right\}.$$

This recursive definition means that the (k)-or-less neighborhood of A at time t is the union, for all the neighbors X reachable by A at time t, of the $(k-1)$-or-less-neighborhoods of the different X at time $t+\mathbf{T}_{A,X}(t)+\delta$. Colloquially, a (k)-neighbor of A is a node which can be reached in k or less hops by a message sent from A at time t, and relayed through any other nodes. As expected, the (1)-or-less-neighborhood is the same thing as the neighborhood given in Definition 2. According to our definition, (k)-or-less-neighborhood has the following straightforward property.

Property 1 *Let $\mathbf{N} = (V, \mathbf{T}(t))$ be a network, then we have:*

$$\mathbf{Ng}_A^{\leq k}(t) \subseteq \mathbf{Ng}_A^{\leq (k+1)}(t).$$

We now define the (k)-neighborhood, which is the set of nodes for which there is a path (in the sense of the previous definition) of length exactly k. It is the set of all the nodes B for which there is a path starting at A at time t and ending at B with k hops, but there is no such path with $k-1$ hops or less.

Definition 6 ((k)-neighborhood). Let $\mathbf{N} = (V, \mathbf{T}(t))$ be a network, $\mathbf{Ng}_A^k$ ((k)-neighborhood of A) is defined by:

$$\mathbf{Ng}_A^k(t) = \left(\mathbf{Ng}_A^{\leq k}(t)\right) \setminus \left(\mathbf{Ng}_A^{\leq k-1}(t)\right).$$

Now, as in the previous subsection, we can formally define what is a protocol which verifies the (k)-or-less-neighborhood relation in our framework.

Definition 7. Let $\mathcal{N}$ be a network family. A protocol $\mathcal{P}$ verifies the (k)-or-less-neighborhood relation in presence of an intruder $\mathcal{I}$ if and only if $\forall \mathbf{N} = (V, \mathbf{T}(t)) \in \mathcal{N}, \forall A, B \in V$, $\nexists tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ such that $END(A, B, t, t_x) \in tr \wedge B \notin \mathbf{Ng}_A^{\leq k}(t)$.

1.4 Example: Authenticated Ranging Protocol

We consider the protocol proposed in [PPH08a]. This protocol aims to verify (1)-neighborhood between two nodes, by using an upper bound on message transfer time between neighbors. We first describe the protocol, then we give a modelling of this protocol, and after that we prove its correctness in our setting.

1.4.1 Description of the Protocol

The idea of the protocol is simple: a message is sent with a nonce, and anyone hearing it replies with that nonce, signed. An Alice-Bob representation of this protocol is given in Figure 1.8.

$$A \xrightarrow{N_A} B$$

$$A \xleftarrow{\{B,N_A\}_{sk_B}} B$$

Fig. 1.8 Authenticated protocol communications diagram, where N_A denotes the nonce generated by A and $\{m\}_{sk_B}$ denotes the message m signed by B.

1.4.2 Protocol Modelling

This protocol is in one way a secured version of the Ping Pong protocol described above. If the resulting message is received before $t_{send} + 2 * t_{max\_emitter} + \delta$, then the emitter is a neighbor (since it is both closer than $t_{max\_emitter}$ and able to communicate with us). This way, the protocol can successfully determine if the node is in communication range or not based on physical property induced by the properties of the nodes. We now give the rules modelling this protocol:

$$(AuthRanging\_1)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}{(tr \cup \{send_\alpha(A, New\_Nonce(), t)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(AuthRanging\_2)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad recv_\alpha(A, N_B, t_1) \qquad t_1 \leq t_2}{(tr \cup \{send_\alpha(A, \langle A, \{N_B\}_{sk_A}\rangle, t_2)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(AuthRanging\_3)\frac{\begin{array}{c} tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad send_\alpha(A, N_A, t_1) \qquad send_\phi(A, N_A, t_{s_A}) \\ recv_\phi(A, \langle B, \{N_A\}_{sk_B}, t_{r_A}) \qquad recv_\alpha(A, \langle B, \{N_A\}_{sk_B}, t_4) \\ t_1 \leq t_{s_A} \leq t_{r_A} \leq t_4 \leq t_5 \qquad t_{r_A} - t_{s_A} \leq (2 * t_{max\_emitter} + \delta) \end{array}}{(tr \cup \{END(A, B, t_{s_A}, t_5)\}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

1.4.3 Authenticated Ranging Protocol Satisfies the Neighborhood Property

In order to keep our proof concise, we assume symmetry in the time needed to send and receive a message, i.e. $\mathbf{T}_{A,B} = \mathbf{T}_{B,A}$ (the $\mathcal{N}_{sym}$ family of networks). We first precise the intruder model we are using, then we show a proof of the protocol. We consider that there is at most one compromised secret key and one compromised node in the whole network. Otherwise, if an intruder has access to a secret key that is not his own (whether it's shared between different nodes, or the intruder has access to a set of them), it is easy to find a trace where an intruder I masquerades as the other intruder J and leads to a $END(A, J)$ event, which by Definition 4 and 7 makes the protocol insecure. Therefore we assume intruders can only use their own secret key.

We show that if $\delta < t_{max\_emitter}$ then we can find an attack, but we also prove that if $\delta \geq t_{max\_emitter}$ then the protocol is correct. Our analysis allows us to deduce sufficient conditions of application for the protocol.

1.4.3.1 Existence of an Attack if $\delta < t_{max\_emitter}$

We consider a network containing three nodes A, B and an intruder I, with constant time of transfer ϵ. The topology of this network is given in Figure 1.9, where there is no link between nodes A and B, and I can communicate with A and B. Applying

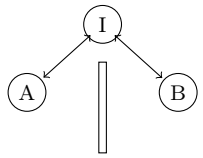

Fig. 1.9 Network of this counter-example

the rules of the authenticated ranging protocol, we can construct the following valid tr in $S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ which contains:

- $send_\alpha(A, N_A, 0)$
- $send_\phi(A, N_A, 0.5\delta)$ (Emission time for A)
- $recv_\phi(I, N_A, 0.5\delta + \epsilon)$
- $recv_\alpha(I, N_A, 1\delta + \epsilon)$
- $send_\alpha(I, N_A, 1\delta + \epsilon)$
- $send_\phi(I, N_A, 1.5\delta + \epsilon)$
- $recv_\phi(B, N_A, 1.5\delta + 2\epsilon)$
- $recv_\alpha(B, N_A, 2\delta + 2\epsilon)$
- $send_\alpha(B, \{B, N_A\}_{sk_B}, 2\delta + 2\epsilon)$
- $send_\phi(B, \{B, N_A\}_{sk_B}, 2.5\delta + 2\epsilon)$
- $recv_\phi(I, \{B, N_A\}_{sk_B}, 2.5\delta + 3\epsilon)$
- $recv_\alpha(I, \{B, N_A\}_{sk_B}, 3\delta + 3\epsilon)$
- $send_\alpha(I, \{B, N_A\}_{sk_B}, 3\delta + 3\epsilon)$
- $send_\phi(I, \{B, N_A\}_{sk_B}, 3.5\delta + 3\epsilon)$
- $recv_\phi(A, \{B, N_A\}_{sk_B}, 3.5\delta + 4\epsilon)$ (Reception time for A)
- $recv_\alpha(A, \{B, N_A\}_{sk_B}, 4\delta + 4\epsilon)$
- $END(A, B, 0.5\delta, 4\delta + 4\epsilon)$

Now, the *END* event could only be generated if $(3.5\delta + 4\epsilon) - (0.5\delta) < 2 * t_{max\_emitter} + \delta$, according to rule $(AuthRanging_3)$. It can be simplified to $\delta + 2 * \epsilon < t_{max\_emitter}$. Since the *END* event claims neighborhood between A and B at time 0.5δ while there is no neighborhood relation between them, the protocol is flawed if $\delta < t_{max\_emitter}$.

1.4.3.2 The protocol is Secure if $\delta \geq t_{max\_emitter}$

In other words we prove that the protocol will never claim there is a neighborhood property between two nodes, when there is actually none. In order to do this, we proceed by contradiction, assuming the protocol is broken and showing a contradiction. We assume the protocol does not satisfy the definition, this means that:

$$\exists tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \ s.t. \ END(A, B, t_{s_A}, t) \in tr \wedge B \notin \mathbf{Ng}_A^1(t_{s_A}).$$

Due to the cryptographic primitives, we are sure that the replied message is forged by B and cannot be modified, since an intruder who is not B would not have access to sk_B. Hence we are sure that the message went through B. Then we compute a lower bound for the running time of the protocol by decomposing it in three phases: from A to B, during B's computations and from B to A.

- From A to B: Following the rules of the protocol, in order to generate $END(A, B, t_{s_A}, t) \in tr$, the following properties must be true :
  - $send_\phi(A, N_A, t_{s_A}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$
  - $recv_\phi(A, \langle B, \{N_A\}_{sk_B}, t_{r_A}) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$
  - $t_{r_A} - t_{s_A} \leq (2 * t_{max\_emitter} + \delta)$

  N_A has been sent by A at t_{s_A}, the time when B receives it is denoted by t_{r_B}. Hence we have that $t_{transfer}(A, B, t_{s_A}) \leq t_{r_B} - t_{s_A}$. Applying lemma 1 we obtain:
  $$B \notin \mathbf{Ng}_A^1(t_{s_A}) \Leftrightarrow t_{transfer}(A, B, t_{s_A}) > t_{max\_emitter}.$$
  We deduce that:
  $$t_{r_B} - t_{s_A} > t_{max\_emitter}$$
  B also has access to N_A at least at time t_{r_B} (and not before).
- During B's computations: Applying the rules, we know that the forward time $(t_{s_B} - t_{r_B})$ is greater or equal to δ.
- From B to A: We have $t_{transfer}(B, A, t_{s_B}) \leq t_{r_A} - t_{s_B}$. Because we are in a symmetrical network and due to assumption 1 and 2, we also have $A \notin \mathbf{Ng}_B^1(t_{s_B})$ then we apply lemma 1 once again regarding this message, and therefore A receives the signed answer at best at time t_{r_A} such that:
  $$A \notin \mathbf{Ng}_B^1(t_{s_B}) \Leftrightarrow t_{r_A} - t_{s_B} > t_{max\_emitter}$$

By summing all these bounds we get:

$$t_{r_A} - t_{s_A} = (t_{r_A} - t_{s_B}) + (t_{s_B} - t_{r_B}) + (t_{r_B} - t_{s_A}) > 2 * t_{max\_emitter} + \delta$$

According to the hypothesis related to the *END* event, we have $t_{r_A} - t_{s_A} \leq 2 * t_{max\_emitter} + \delta$ which leads us to a contradiction.

1.5 (k)-or-less-Neighbors Discovery Protocol

Assuming that we have a secure (1)-neighbor discovery protocol, we propose a (k)-or-less-neighbor discovery protocol based on it. This protocols aims to construct the (k)-or-less-neighborhood, by using the knowledge each node has about its neighborhood. For this protocol, we need to consider several *END* events, each being a stepping stone towards a final conclusion. For instance $END^k(A, B, t_{prop}, t)$ expresses the (k)-or-less neighborhood property between A and B at time t_{prop}. We notice that definitions of how a protocol verifies a property follows the same idea as before: there should not exist a trace where an *END* contradicts the physical property it claims. We call this protocol the *Sharek* protocol.

1.5.1 Description of the protocol

In the first phase of the protocol, each node floods the network with its (1)-neighborhood. After this, no more communications happen.

Then, all the nodes try to map their (2)-or-less-neighborhoods based on **two or more of their** (1)**-neighbors claiming neighborhood to another node**. After that, each node continues by computing its upper neighborhood based on its previous deductions. We do not require that each node knows its whole (1)-neighborhood, but only a subset of it. We model this with the $View$ function, which is used in the rules of the protocol.

This protocol is sensitive to the number of intruders, this simple version works only when there is only one intruder in the network. This way, the single intruder can not create false conclusions by lying, since there will not be the same claim from another (necessarily legitimate) node unless it was true in the first place. If we increase the number of approving nodes in the deduction, then we can tolerate more intruders.

There is a small improvement we stacked on top of this : if two nodes claim neighborhood with a third one, but are not at the same level of neighborhood with the center, then the third one will be accepted at the highest level. Since the $(k-1)$-or-less neighborhood is included in the (k)-or-less neighborhood, we can choose the most conservative option and still satisfy the conditions for the verification of neighborhood properties.

The security proof shows that if the protocol ends by reaching a conclusion, the results are secure. We also assume one of the two following assumptions:

- Nodes have synchronized clocks. This is required by the protocol, which makes nodes use the timestamped data sent by their neighbors. If honest nodes send erroneous times, then the intruder can easily leverage this and trick the protocol into a false conclusion. In the rest, we consider this assumption.
- Nodes are static. *i.e.* $\mathcal{N}_{still}$ as defined above. Then, the neighborhoods will not change over time, and the timestamps become trivially useless. It is easy to infer a proof with this assumption from the previous case.

We model the knowledge of a part of the (1)-neighbors by the View function, which returns the part of the (1)-neighbors that the node knows. A high level description of Sharek Protocol is given in Figure 1.10, where the double arrow symbolize the flooding by repeated broadcasts.

$$\forall X \xrightarrow{\{View(\mathbf{Ng}^1_X(t_{prop}))\}_{sk_X}} *$$

Fig. 1.10 Sharek protocol communications diagram

1.5.2 Rules of Sharek Protocol

We give the few rules modeling our protocol:

$$(Sharek\_begin)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}{tr.send_\alpha(A, \{\langle t_{prop}, A, View(A, \mathbf{Ng}^1_A)\rangle\}_{sk_A}, t) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(Sharek\_basis)\frac{tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \qquad B \in View(A, \mathbf{Ng}^1_A(t_{prop})) \qquad t_{prop} \leq t}{tr.END^1(A, B, t_{prop}, t) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(Sharek\_forward)\frac{\begin{array}{c} tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \\ recv_\alpha(A, \{\langle t_{prop}, B, \{N_1, ..., N_k\}\rangle\}_{sk_B}, t_1) \in tr \\ t_1 \leq t_2 \end{array}}{tr.send_\alpha(A, \{\langle t_{prop}, B, \{N_1, ..., N_k\}\rangle\}_{sk_B}, t_2) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

$$(Sharek\_makestep)\frac{\begin{array}{c} tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \\ recv_\alpha(A, \{\langle t_{prop}, B, \{D, ...\}\rangle\}_{sk_B}, t_1) \in tr \\ recv_\alpha(A, \{\langle t_{prop}, C, \{D, ...\}\rangle\}_{sk_C}, t_2) \in tr \\ END^b(A, B, t_{prop}, t_b) \in tr \\ END^c(A, C, t_{prop}, t_c) \in tr \\ \forall t_x \in \{t_1, t_2, t_b, t_c\}, t_x \leq t_d \\ (max(b, c) + 1) \leq k \end{array}}{tr.END^{max(b,c)+1}(A, D, t_{prop}, t_d) \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}}$$

1.5.3 Security of Sharek

In order to prove that the protocol verifies the (k)-or-less-neighborhood property, we proceed with a proof by induction. First we prove that our protocol verifies (1)-neighborhood property in Lemma 2, then we assume Sharek protocol verifies all (i)-neighborhood property for $i \leq k$ and we show that it verifies $(k+1)$-neighborhood property in Lemma 3.

We define $\mathcal{N}_{1,sym}$ as the family of networks where there is only a single intruder, and where all the connections are symmetric.

Lemma 2. *The protocol verifies the* (1)*-neighborhood property over* $\mathcal{N}_{1,sym}$*:*

$$\forall \mathbf{N} \in \mathcal{N}_{1,sym}, \nexists tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \text{ s.t. } END^1(A, B, t_{prop}, t) \in tr \wedge B \notin \mathbf{Ng}^1_A(t_{prop})$$

Proof. We assume that $\exists tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ *s.t.* $END^1(A, B, t_{prop}, t) \in tr \wedge B \notin \mathbf{Ng}^1_A(t_{prop})$. The only way we can generate a $END^1(A, B, t_{prop}, t)$ event in a trace $tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ is using $(Sharek\_basis)$. This rule requires $B \in View(A, \mathbf{Ng}^1_A(t_{prop}))$. The $View$ function, by definition, returns a subset of $\mathbf{Ng}^1_A(t_{prop})$: therefore, there is a contradiction. □

Lemma 3. *If the protocol verifies all the i-or-less neighborhood properties,* $0 < i \leq k$ *over* $\mathcal{N}_{1,sym}$*, then it verifies the* $(k+1)$*-neighborhood property over* $\mathcal{N}_{1,sym}$*. More formally it means that:*

$$\forall \mathbf{N} \in \mathcal{N}_{1,sym}, \forall i \leq k, \nexists tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \ s.t. \ END^i(A, D, t_{prop}, t) \in tr \wedge D \notin \mathbf{Ng}_A^{\leq i}(t_{prop})$$

$$\Rightarrow$$

$$\forall \mathbf{N} \in \mathcal{N}_{1,sym}, \nexists tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}} \ s.t. \ END^{k+1}(A, D, t_{prop}, t) \in tr \wedge D \notin \mathbf{Ng}_A^{\leq k+1}(t_{prop})$$

Proof. We do a proof by contradiction. We assume that there is a $\mathbf{N} \in \mathcal{N}_{1,sym}$ and a $tr \in S_{\mathbf{N},\mathcal{I},\mathcal{P}}$ such that $END^{k+1}(A, D, t_{prop}, t) \in tr \wedge D \notin \mathbf{Ng}_A^{\leq k+1}(t_{prop})$. Since $END^{k+1}(A, D, t_{prop}, t) \in tr$ then it can only be build from ($Sharek\_makestep$), meaning that we must have all the hypothesis true, in particular:

- $recv_\alpha(A, \{\langle t_{prop}, B, \{D, ...\}\rangle\}_{sk_B}, t_1) \in tr$
- $recv_\alpha(A, \{\langle t_{prop}, C, \{D, ...\}\rangle\}_{sk_C}, t_2) \in tr$
- $END^k(A, B, t_{prop}, t_x) \in tr$
- $END^i(A, C, t_{prop}, t_y) \in tr, i \leq k$

We apply the induction hypothesis on the two last items. We obtain that $B \in \mathbf{Ng}_A^{\leq k}(t_{prop})$ and $C \in \mathbf{Ng}_A^{\leq i}(t_{prop}) \subseteq \mathbf{Ng}_A^{\leq k}(t_{prop})$ (and due to property 1). By hypothesis we know that $D \notin \mathbf{Ng}_A^{\leq k+1}(t_{prop})$, which is equivalent to

$$\left(\nexists X \in V \ s.t. \ X \in \mathbf{Ng}_A^{\leq k}(t_{prop}) \wedge D \in \mathbf{Ng}_X^1(t_{prop} + t_X)\right).$$

We deduce that $D \notin \mathbf{Ng}_X^1(t_{prop} + t_X)$ for $X \in \{B, C\}$ with t_X the minimal time needed to forward a message from A to X at t_{prop}. Both messages contained in the receive events are signed by their pretended emitters, respectively B and C. We can then conclude that both B and C crafted their respective claims of neighborhood with D. But C and B claims in these messages that D is in their (1)-neighborhood. Since nodes can not lie about their neighborhoods using rules in $\mathcal{P}$, this means that both their messages were built by ($Intrude\_forge$), and as both messages are forged by their pretended emitter, we can conclude that both B and C are intruders. This is in contradiction with our initial hypothesis that the network consists of only one intruder. □

1.6 Conclusion

We have proposed a way of modeling physical properties in a wireless network in order to verify protocols in the presence of intruders, taking into account time and movement of nodes. We focused on the protocols which discover the distance between nodes, from a single hop (usually called neighborhood property) to an arbitrary number of them ((k)-neighborhood). After introducing the model, we have applied the model to two protocols and proved that one is correct under some assumptions about the network, and the intruder. Finally we propose the Sharek protocol, which securely discover (k)-neighbors based on the knowledge of the (1)-neighborhood, in presence of one intruder. We can generalize this protocol in order to be resistant to several intruders. We also provide a formal proof of the security of the Sharek protocol as a third example of an application using our formal model. This formal modelling of the (k)-neighborhood is the first step toward an automatic tool for verifying neighborhood discovery protocols.

References

[ABK+11] Gildas Avoine, Muhammed Ali Bingöl, Süleyman Kardaş, Cédric Lauradoux, and Benjamin Martin. A framework for analyzing rfid distance bounding protocols. *J. Comput. Secur.*, 19:289–317, April 2011.

[And01] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[BC94] S. Brands and D. Chaum. Distance-bounding protocols. In *Advances in Cryptology (EUROCRYPT'93)*, pages 344–359. Springer, 1994.

[BCM11] David Basin, Cas Cremers, and Catherine Meadows. *Model Checking Security Protocols*, chapter 24. Springer, 2011. To appear.

[BMSU01] Prosenjit Bose, Pat Morin, Ivan StojmenoviÄĞ, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7:609–616, 2001. 10.1023/A:1012319418150.

[CH05] Srdjan Capkun and Jean-Pierre Hubaux. Secure positioning of wireless devices with application to sensor networks. In *IEEE INFOCOM*. IEEE Journal on Selected Areas in Communications: Special Issue on Security in Wireless Ad Hoc Networks, 2005.

[CRC11] Cas Cremers, Kasper Bonne Rasmussen, and Srdjan Capkun. Distance hijacking attacks on distance bounding protocols. Cryptology ePrint Archive, Report 2011/129, 2011. `http://eprint.iacr.org/`.

[DB97] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Communications, 1997. ICC 97 Montreal, 'Towards the Knowledge Millennium'. 1997 IEEE International Conference on*, volume 1, pages 376 –380 vol.1, jun 1997.

[DKPR11] J. Du, E. Kranakis, O. Morales Ponce, and S. Rajsbaum. Neighbor discovery in a sensor network with directional antennae. In *Algosensors*, Saarbruecken, Germany, September 2011.

[DY81] D. Dolev and A.C. Yao. On the security of public key protocols. In *Proc. of the 22nd Symp. on Foundations of ComputerScience*, pages 350–357. IEEE Computer Society Press, 1981.

[DY83] D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.

[HK05] G.P. Hancke and M.G. Kuhn. An rfid distance bounding protocol. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 67–73. IEEE, 2005.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[HPJ03] Y. C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 3, pages 1976–1986 vol.3, 2003.

[Low96] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96)*, volume 1055, pages 147–166. Springer-Verlag, Berlin Germany, march 1996. Also in Software Concepts and Tools, 17:93-102, 1996.

[Low97] G. Lowe. Casper: A compiler for the analysis of security protocols. In *Proc. of 10th Computer Security Foundations Workshop (CSFW'97)*. IEEE Computer Society Press, 1997. Also in Journal of Computer Security, Volume 6, pages 53-84, 1998.

[MPP+06] Catherine Meadows, Radha Poovendran, Dusko Pavlovic, LiWu Chang, and Paul Syverson. Distance bounding protocols: Authentication logic analysis and collusion attacks. In *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks, edited volume, Springer, Nov. 2006*, Nov 2006.

[NPW02] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[NS78] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communication of the ACM*, 21(12):993–999, 1978.

[Pau98] L.C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of computer security*, 6(1-2):85–128, 1998.

[PCTS02] Adrian Perrig, Ran Canetti, J. D. Tygar, and Dawn Song. The tesla broadcast authentication protocol. *RSA CryptoBytes*, 5, 2002.

[PPH07] Marcin Poturalski, Panos Papadimitratos, and Jean-Pierre Hubaux. Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possiblity. Technical report, EPFL, 2007.

[PPH08a] M. Poturalski, P. Papadimitratos, and J.P. Hubaux. Secure neighbor discovery in wireless networks: formal investigation of possibility. In *Proceedings of the 2008 ACM*

symposium on Information, computer and communications security, pages 189–200. ACM, 2008.

[PPH08b] Marcin Poturalski, Panos Papadimitratos, and Jean-Pierre Hubaux. Towards Provable Secure Neighbor Discovery in Wireless Networks. In *The 6th ACM Workshop on Formal Methods in Security Engineering*, pages 31–42, Alexandria, VA, 2008. ACM.

[PPS+08] Panagiotis (Panos) Papadimitratos, Marcin Poturalski, Patrick Schaller, Pascal Lafourcade, David Basin, Srdjan Capkun, and Jean-Pierre Hubaux. Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networking. *IEEE Communications Magazine*, 46(2), 2008.

[RSG+00] P.Y.A. Ryan, S.A. Schneider, M.H. Goldsmith, G. Lowe, and A.W. Roscoe. *The modelling and analysis of security protocols: the CSP approach*. Addison-Wesley, 2000.

[Sch96] S.A. Schneider. Security properties and CSP. In *Proc. of the Symposium on Security and Privacy*, pages 174–187. IEEE Computer Society Press, 1996.

[SPR+09] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.P. Hubaux. A practical secure neighbor verification protocol for wireless sensor networks. In *Proceedings of the second ACM conference on Wireless network security*, pages 193–200. ACM, 2009.

[SSBC09] Patrick Schaller, Benedikt Schmidt, David Basin, and Srdjan Capkun. Modeling and verifying physical properties of security protocols for wireless networks. In *Proceedings of the IEEE Computer Security Foundations Symposium (CSF)*, pages 109–123. IEEE, 2009.

[TSG10] F. Javier Thayer, Vipin Swarup, and Joshua D. Guttman. Metric strand spaces for locale authentication protocols. In Masakatsu Nishigaki, Audun Jøsang, Yuko Murayama, and Stephen Marsh, editors, *Trust Management IV - 4th IFIP WG 11.11 International Conference, IFIPTM 2010, Morioka, Japan, June 16-18, 2010. Proceedings*, volume 321 of *IFIP Conference Proceedings*, pages 79–94. Springer, 2010.

[VKT05] S. Vasudevan, J. Kurose, and D. Towsley. On neighbor discovery in wireless networks with directional antennas. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2502–2512. IEEE, 2005.

[WBRW07] J. Wilson, V. Bhargava, A. Redfern, and P. Wright. A wireless sensor network and incident command interface for urban firefighting. In *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*, pages 1–7. IEEE, 2007.

[WL99] Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '99, pages 7–14, New York, NY, USA, 1999. ACM.