Formal Analysis of Secure Neighbor Discovery in Wireless Networks

Marcin Poturalski, *Student Member*, *IEEE*, Panos Papadimitratos, *Member*, *IEEE*, and Jean-Pierre Hubaux, *Fellow*, *IEEE*

Abstract—We develop a formal framework for the analysis of security protocols in wireless networks. The framework captures characteristics necessary to reason about neighbor discovery protocols, such as the neighbor relation, device location, and message propagation time. We use this framework to establish general results about the possibility of neighbor discovery. In particular, we show that time-based protocols cannot in general provide secure neighbor discovery. Given this insight, we also use the framework to prove the security of four concrete neighbor discovery protocols, including two novel time-and-location-based protocols. We mechanize the model and some proofs in the theorem prover Isabelle.

Index Terms—Neighbor discovery, relay attack, formal verification, distance bounding

1 Introduction

Wireless communications are flexible: a device with a wireless interface can start communicating with another device, an access point, or a base station, almost instantly, without setting up a cable connection. As a consequence, wireless connections are frequently established, making *discovering* devices available for direct communication an indispensable element of wireless networks. However, due to the open nature of wireless communication, *neighbor discovery (ND)* is easy to abuse: An adversary can convince a device into falsely believing that another device is its neighbor. The adversary can then use these false neighbor links to disrupt the applications and services that use ND as a building block.

The canonical example of such an attack comes from routing in wireless ad hoc networks, such as sensor networks, or multihop smart phone networks [1]. An adversary can abuse ND by mounting a *relay attack* (also known as a *wormhole attack* [15], [21]): Equipped with two devices connected by a fast out-of-band link, the adversary relays (without modification) any message overheard by one device to the other side of the wormhole (Fig. 1). This can create shortcuts across the network the routing protocols are attracted to, with a significant portion of the traffic routed through the wormhole. The adversary can then eavesdrop, modify, or simply suddenly start dropping messages, causing a denial-of-service.

Securing ND is clearly of utter importance. It has therefore attracted considerable attention from the research community and a number of secure ND protocols have been proposed. However, in the vast majority of cases, such protocols have been only argued secure in an informal manner. History provides many examples of flaws in security protocols that can be overlooked by such arguments. It is, hence, only natural to strive for the strong security guarantees offered by *formal methods*. This is the main goal of this paper: formal analysis of secure ND.

To this end, we develop a formal framework modeling wireless communications. Our framework captures wireless communication aspects necessary to reason about ND protocols, notably:

- the neighbor relation,
- device location,
- message propagation time, and
- message transmission time (not instantaneous).

Such aspects were abstracted away by "traditional" formal approaches designed for verification of security protocol in Internet-like environments.

Our framework allows us to obtain two types of results. First, under what assumptions a given class of protocols can or cannot provide secure ND? In particular, we prove that a general class of *time-based protocols* cannot provide ND if the adversary can relay messages with a delay below a threshold determined by the desired communication range of the ND protocol. Second, which concrete protocols can be proven secure? With our framework, we prove the security of four such protocols, two time-based ones and two in a different general class of *time-and-location-based* protocols. Such proofs greatly increase the confidence in the protocols and they serve as secure ND possibility results for the respective classes of protocols. Finally, we mechanize our framework and a number of protocol security proofs in the theorem prover Isabelle [19]. This provides an additional level of assurance.

Paper outline. In Section 2, we define the framework and the ND specification. In Section 3, we sketch the time-based protocol impossibility. In Section 4, we define the protocols a refined ND specification, and present the security analysis results. We also give an overview of the Isabelle/HOL

- *M. Poturalski and J.-P. Hubaux are with the Swiss Federal Institute of Technology in Lausanne, Lausanne, Switzerland.*
- *P. Papadimitratos is with the Royal Institute of Technology, Osquldas väg 10, Stockholm SE-100 44, Sweden.*

Manuscript received 3 Aug. 2011; revised 1 July 2012; accepted 10 Jan. 2013; published online 13 Mar. 2013.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-2011-08-0188. Digital Object Identifier no. 10.1109/TDSC.2013.17.

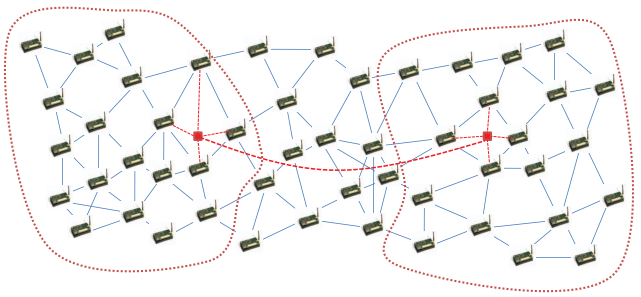

Fig. 1. Wormhole attack in a wireless sensor network: The adversary relays packets between two adversarial devices (squares), creating false neighbor links. As a result, traffic between the left and the right parts of the network (circled) traverses the wormhole (thick dashed curve).

mechanization. We discuss the assumptions and compare the protocols in Section 5. In Section 6, we discuss the related work and open problems, and we conclude the paper in Section 7.

2 System Model

The basic wireless network entities, *nodes*, are processes running on computational platforms equipped with transceivers communicating over a wireless channel. We assume that nodes have synchronized clocks (although not all protocols we consider in this paper make use of this assumption) and are static (not mobile). Nodes either follow the implemented system functionality, in which case we denote them as *correct* or *honest*, or they are under the control of an adversary, in which case we denote them as *adversarial* nodes. Adversarial nodes can behave in an arbitrary fashion, also acting as correct nodes or lying dormant for any period of time.

To capture the inherent characteristics of ND in wireless networks, we model communication at the physical layer rather than at higher layers (data link, network, or application). For simplicity, correct nodes are assumed to use a single wireless channel, but we do not require them to have equal transmission power and receiver sensitivity. Adversarial nodes can communicate across the wireless channel used by correct nodes, but they can also communicate across a dedicated *adversarial channel* imperceptible to correct nodes.

Our system model comprises: 1) a *setting* $\mathcal{S}$ that describes the type (correct or adversarial) of nodes, their location, and the state of the wireless channel; 2) a *protocol model* $\mathcal{P}$ that determines the behavior of correct nodes; and 3) an *adversary model* $\mathcal{A}$ that establishes the capabilities of adversarial nodes.

We assume that looking at the system at any point in time reveals one or more phenomena. We are interested in those relevant to the wireless communication and the system at hand and thus to our analysis. We denote these phenomena, associated with nodes, as *events* (Definition 3). Then, we model the system evolution over time using the notion of *trace*, i.e., a set of events (Definition 4), in particular *feasible* traces that satisfy constraints specified by $\mathcal{S}$ (correspondence between wireless sending and receiving of messages), $\mathcal{P}$ (correct nodes follow the protocol), and $\mathcal{A}$ (adversarial nodes behave according to their capabilities). The constraints are defined by logical formulas we call *rules*.

The model presented in the rest of this section is used in Section 3 for the impossibility result. To reason about the security of concrete protocols, we extend the model in Section 4. In Section 5.1, we explain and justify all simplifying modeling assumptions, made to keep the model tractable.

2.1 System Parameters

The technologies used by correct and adversarial nodes determine our model parameters:

- $\mathbf{v} \in \mathbb{R}_{>0}$, the *signal propagation speed*, defining how fast messages propagate across the wireless channel, determined by the communication technology,
- $\mathbf{v}_{\text{adv}} \geq \mathbf{v}$, the *information propagation speed* over the *adversarial channel*; as $\mathbf{v}_{\text{adv}} \geq \mathbf{v}$ this is also the maximum speed at which information can propagate,
- $\Delta_{\text{relay}} \in \mathbb{R}$, the *minimum relaying delay* introduced by a node when relaying a message; this delay is due to processing exclusively, it does not include propagation time or any other delay.
- $\mathbb{M}$, the *message space*; we keep the message space unspecified for the impossibility result in Section 3; we provide a concrete message space when we talk about specific protocols in Section 4.
- $|.| : \mathbb{M} \to \mathbb{R}_{>0}$, the *message duration* function, determines the message transmission time.

Further, $\mathbb{W}$ denotes the set of unique *node identifiers*, which for simplicity we will consider equivalent with the nodes themselves.[1]

2.2 Settings

A setting describes the type and location of nodes, and how the state of the wireless channel changes over time.

Definition 1. *A* setting $\mathcal{S}$ *is a tuple* $\langle V, loc, type, link, nlos \rangle$, *where*

- $V \subset \mathbb{W}$ *is a finite set of nodes. An ordered pair* $(A, B) \in V^2$ *is called a* link.
- $loc : V \to \mathbb{R}^3$ *is the node* location *function. As we assume nodes are not mobile, this function does not depend on time. We define* $dist : V^2 \to \mathbb{R}_{\geq 0}$ *as* $dist(A, B) = d(loc(A), loc(B))$, *where* d *is the euclidean distance in* $\mathbb{R}^3$. *We require the* loc *function to be injective, so that no two nodes share the same location. Thus,* $dist(A, B) > 0$ *for* $A \neq B$.
- $type : V \to \{correct, adversarial\}$ *is the* type *function; it defines which nodes are* correct *and which are* adversarial. *This function does not depend on time, as we assume that the adversary does not corrupt new nodes during the system execution. We denote* $V_{\text{cor}} = type^{-1}(\{correct\})$ *and* $V_{\text{adv}} = type^{-1}(\{adversarial\})$.
- $link : V^2 \times \mathbb{R}_{\geq 0} \to \{up, down\}$ *is the* link state *function. Accordingly, we say that at a given time* $t \geq 0$, *a* link $(A, B) \in V^2$ *is* up *(denoted* $link(A \to B, t)$*) or* down *(denoted* $link(A \not\to B, t)$*). We use*

1. Although this implies that every node is assigned a single identifier, it does not prevent an adversarial node from using (in the messages in sends) any identifier.

$$\text{S1}\quad \forall A \in V, t \in \mathbb{R}_{\geq 0}, m \in \mathbb{M}.\ \ \mathsf{Receive}(A;t;m) \in \theta \implies \exists B \in V.\ \ link(B{\rightarrow}A, [t, t+|m|]) \wedge (\mathsf{Bcast}(B; t-(dist(A,B)+nlos(A,B))\mathbf{v}^{-1}; m) \in \theta)$$

$$\text{S2}\quad \forall A, B \in V, t \in \mathbb{R}_{\geq 0}, m \in \mathbb{M}.\ \ \mathsf{Bcast}(B; t-(dist(A,B)+nlos(A,B))\mathbf{v}^{-1}; m) \in \theta \wedge link(B{\rightarrow}A, [t, t+|m|])) \implies \mathsf{Receive}(A;t;m) \in \theta$$

$$\text{S3}\quad \forall A \in \mathbb{V}, t \in \mathbb{R}_{\geq 0}, m \in \mathbb{M}.\ \ (\mathsf{Receive}(A;t;m) \in \theta \vee \mathsf{Bcast}(A;t;m) \in \theta \implies A \in V)$$

$$\text{P1}\quad \forall A \in V_{\text{cor}}, t \in \mathbb{R}_{\geq 0}, m \in \mathbb{M}.\ \ \mathsf{Bcast}(A;t;m) \in \theta \implies \mathsf{Bcast}(m) \in \mathcal{P}(\theta||_{A,t})$$

$$\text{P2}\quad \forall A \in V_{\text{cor}}, t, t' \in \mathbb{R}_{\geq 0}, B, C \in \mathbb{V}.\ \ \mathsf{Neighbor}(A;t;B,C,t') \in \theta \implies \mathsf{Neighbor}(B,C,t') \in \mathcal{P}(\theta||_{A,t})$$

$$\text{P3}\quad \forall A \in V_{\text{cor}}.\ \ \forall t \in E_A.\ \ \epsilon \in \mathcal{P}(\theta||_{A,t})$$
where $E_A = \mathbb{R}_{\geq 0} \setminus start(\theta|_A \cap I)$
and $I = \{\mathsf{Bcast}(t;m) \mid m \in \mathbb{M}, t \in \mathbb{R}_{\geq 0}\} \cup \{\mathsf{Neighbor}(t;B,C,t') \mid B, C \in \mathbb{V}, t, t' \in \mathbb{R}_{\geq 0}\}$

$$\text{A1}\quad \forall A \in V_{\text{adv}}, t \in \mathbb{R}_{\geq 0}, m \in \mathbb{M}.\ \ \mathsf{Bcast}(A;t;m) \in \theta \implies \exists B \in V_{\text{adv}}, \delta \geq \Delta_{\text{relay}} + dist(B,A)\mathbf{v}_{\text{adv}}^{-1}.\ \ \mathsf{Receive}(B; t-\delta; m) \in \theta$$

Fig. 2. Setting-feasibility rules, protocol-feasibility rules for protocol model $\mathcal{P}$, and adversary-feasibility rule for adversary model $\mathcal{A}_{\Delta_{\text{relay}}}$.

abbreviations $link(A \leftrightarrow B, t) =_{\text{def}} link(A \rightarrow B, t) \wedge link(B \rightarrow A, t)$ *and* $link(A \not\leftrightarrow B, t) =_{\text{def}} link(A \not\rightarrow B, t) \wedge link(B \not\rightarrow A, t)$. *We extend the "*$link(A \rightarrow B, t)$*" notation from single time points to sets as follows:* $link(A \rightarrow B, T) =_{\text{def}} \forall t \in T \in link(A \rightarrow B, t)$. *We establish the convention* $link(A \not\rightarrow A, \mathbb{R}_{\geq 0})$.

- $nlos : V^2 \rightarrow \mathbb{R}_{\geq 0}$ *is the* non-line-of-sight delay (NLOS) *function. If two nodes* A *and* B *can communicate over a line of sight, then* $nlos(A,B) = 0$. *Otherwise,* $nlos(A,B)$ *specifies the additional distance that the signal has to propagate compared to line-of-sight propagation* $dist(A,B)$. *We assume this function is symmetric, because of reciprocity of wireless links.*

We denote the set of all settings by $\mathbb{S}$.

We model the ability to communicate directly, without the intervention or "assistance" of relays, by a link being up, thus the following definition:

Definition 2. *Node* A is a neighbor *of node* B *in setting* $\mathcal{S}$ *at time* t, *if* $link(A \rightarrow B, t)$. *If* $link(A \leftrightarrow B, t)$, *nodes* A *and* B are neighbors *at time* t.

For simplicity of presentation, we use "$link(A \rightarrow B, t)$" to denote the neighbor relation and the link relation.

2.3 Events and Traces

Events relate to the wireless communication and the ND protocol operation. Each event is primarily associated with (essentially, takes place at) a node we call the *active* node.

Definition 3. *An* event *is one of the following terms:*

- $\mathsf{Receive}(A;t;m)$
- $\mathsf{Bcast}(A;t;m)$
- $\mathsf{Neighbor}(A;t;B,C,t')$
- $\mathsf{Fresh}(A;t;n)$
- $\mathsf{NDstart}(A;t)$
- $\mathsf{NDstart}(A;t;B)$

where $A \in \mathbb{W}$ *is the* active node, $t \in \mathbb{R}_{\geq 0}$ *is the event* start time, *denoted by* $start(.)$, *and* $m \in \mathbb{M}$ *is the transmitted/received message,* $n \in$ Nonces *is a nonce,* $B, C \in \mathbb{W}$ *are nodes, and* $t' \in \mathbb{R}_{\geq 0}$ *is a time instant.*

The first two events are related to communication on the physical layer: Receive represents message reception, and Bcast represents message transmission.

Neighbor can be thought of as an internal outcome of a ND protocol (possibly reported to some higher layer): Node A declares that B is a neighbor of C at time t'. Having t' a single point in time is for simplicity only, and we could easily generalize to arbitrary sets.

Fresh is used to declare that nonce n is (freshly) generated by A at time t or, in other words, that it was not sent before t. With NDstart, node A declares that an instance of a ND protocol has been initialized: either with a specific node B or with all neighbors.

We use the notion of *trace* to model an execution of the system.

Definition 4. *A* trace θ *is a set of events.*

We denote the set of all traces by Θ. Given a setting $\mathcal{S}$, a protocol $\mathcal{P}$, and an adversary $\mathcal{A}$, we denote the set of traces feasible with respect to $\mathcal{S}$ by $\Theta_{\mathcal{S}}$, the set of those feasible with respect to $\mathcal{S}$ and $\mathcal{P}$ by $\Theta_{\mathcal{S},\mathcal{P}}$, and with respect to $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{A}$ by $\Theta_{\mathcal{S},\mathcal{P},\mathcal{A}}$.

2.4 Setting-Feasible Traces

The feasibility of a trace θ with respect to a setting $\mathcal{S} = \langle V, loc, type, link, nlos \rangle$ ensures a causal and strict time relation between send and receive events; it is formally defined by rules S1-S3 (Fig. 2). Rule S1 ensures that every message that is received was previously sent. Dually, rule S2 ensures that a transmitted message is received by all nodes enabled to do so by the link relation. In other words, communication is causal (a receive is always preceded by a sent), and reliable *as long as the link is up*. Unreliability, expected and common in wireless communications, is modeled by the state of the link being *down*. Furthermore, these rules introduce a strict time relation between events, reflecting the propagation delay from A to B, across the channel, with speed $\mathbf{v}$: $(dist(A,B) + nlos(A,B))\mathbf{v}^{-1}$. Rule S3 is a technical one: It ensures that no communication events are performed by nodes not present in setting $\mathcal{S}$.

2.5 Protocol-Feasible Traces

Intuitively, a trace is feasible with respect to protocol $\mathcal{P}$ if correct nodes behave according to a particular protocol $\mathcal{P}$. To formalize this, we first define the notion of a local view.

A trace is essentially a *global view* of the system execution. To describe what a node observes during a system execution, we use the notion of *local view*, primarily comprising a *local trace* composed of *local events*. We define these next. For simplicity, we ignore the NDstart and Fresh events, as they are of no consequence for the impossibility result.

$$\begin{aligned}\theta|_{A,t} = &\{\mathsf{Bcast}(t_1;m) \mid t_1 < t \ \wedge\ \mathsf{Bcast}(A;t_1;m) \in \theta\} \cup \\ &\{\mathsf{Receive}(t_1;m) \mid t_1 + |m| < t \ \wedge\ \mathsf{Receive}(A;t_1;m) \in \theta\} \cup \\ &\{\mathsf{Neighbor}(t_1;B,C,t') \mid t_1 < t \ \wedge\ \mathsf{Neighbor}(A;t_1;B,C,t') \in \theta\}\end{aligned}$$

Fig. 3. Local trace (Definition 6).

Definition 5. A local event *is one of the terms:*

- $\mathsf{Bcast}(t;m)$,
- $\mathsf{Receive}(t;m)$, *or*
- $\mathsf{Neighbor}(t;B,C,t')$,

where $B,C \in \mathbb{W}$, $m \in \mathbb{M}$, $t,t' \in \mathbb{R}_{\geq 0}$. *For a local event* e, $start(e)$ *is defined as in Definition 3.*

Definition 6. A local trace *is a set of local events. Given a node identifier* $A \in \mathbb{W}$, *time* $t \geq 0$, *and trace* $\theta \in \Theta$, *we calculate the* local trace of node A at time t in trace θ, *denoted* $\theta|_{A,t}$ *as shown in Fig. 3. We call* $\theta|_{A,\infty}$ *a* complete local trace *of* A *in* θ *and denote it shortly* $\theta|_A$.

We identify two variants of the local view notion: a *T-local view*, as the basis for defining the class of time-based protocols, and a *TL-local view*, used to define the class of time- and location-based protocols.

Definition 7. *Given a trace* θ, *a* T-local view of node A at time t in θ *is a tuple* $\langle A,t,\theta|_{A,t}\rangle$; *we denote it* $\theta\|_{A,t}$.

Definition 8. *Given a trace* θ *and a setting* $\mathcal{S}$, *a* TL-local view of node A at time t in θ *is a tuple* $\langle A,t,loc(A),\theta|_{A,t}\rangle$; *we denote it* $\theta\|_{\mathcal{S},A,t}$, *or* $\theta\|_{A,t}$ *as the setting* $\mathcal{S}$ *is clear from the context.*

Note that $\mathcal{S}$ is part of Definition 8 as the location of node A is defined only within a specific setting. With the notion of the local view in hand, we can proceed with the definition of a protocol model. This definition captures the property of protocols essential to our investigation: the fact that protocol behavior depends *exclusively* on the local view of the node executing the protocol. To express this, we model a protocol as a function from the local view to a set of actions. This leads to an overapproximation, as we allow for protocols that are, for example, not computable. However, this is fine for the purpose of the impossibility result.

Definition 9. A T(TL)-protocol model $\mathcal{P}$ *is a function which given a T(TL)-local view* $\theta\|_{A,t}$, *determines a finite, nonempty set of* actions; *an* action *is one of the terms:* ϵ, $\mathrm{Bcast}(m)$ *or* $\mathsf{Neighbor}(B,C,t')$, *where* $m \in \mathbb{M}, B,C \in \mathbb{W}, t' \in \mathbb{R}_{\geq 0}$.

The interpretation of Bcast and Neighbor actions is natural. The ϵ action means that the node does not execute an event, with the exception of possible Receive event(s). Note that modeling the protocol output (i.e., the protocol model codomain) as a family of *sets of actions* allows for nondeterministic protocols.

The feasibility of a trace θ with respect to a protocol model $\mathcal{P}$ ensures that all correct nodes follow the protocol; it is formally defined by rules P1-P3 (Fig. 2). Rules P1 and P2 ensure that the Bcast and Neighbor actions taken by a node are allowed by the protocol. Rule P3, with E_A the set of all time instances in θ when no event other than Receive happens at node A, ensures that the protocol allows for a node perform no action.

Note that our definition of a protocol model only requires that the behavior of the protocol is determined by the local view. This is much broader than a possible alternative approach, in which a protocol is modeled by a Turing machine. But as our definition is an overapproximation, the impossibility result remains valid for more realistic protocol models.

2.6 Adversary-Feasible Traces

For the purpose of the impossibility result, we consider first a relatively limited adversary, that is only capable of relaying messages. Note that a weak adversary model strengthens the impossibility result. We denote this model as $\mathcal{A}_{\Delta_{\text{relay}}}$, with the $\Delta_{\text{relay}} > 0$ parameter the minimum relaying delay introduced by an adversarial node; this delay is due to processing exclusively, and it does not include propagation or transmission time.

Formally, the feasibility of trace θ with respect to $\mathcal{A}_{\Delta_{\text{relay}}}$ is defined by rule A1 in Fig. 2: Every message sent by an adversarial node is necessarily a replay of a message m that either this or another adversarial node received. In addition, the delay between receiving m and resending it, or more precisely the difference between the start times of the corresponding events, needs to be at least Δ_{relay}, plus the propagation delay across the adversary channel (in case another adversarial node received the relayed message). This condition reflects the structure of the adversarial channel: Any two adversarial nodes can establish direct communication.

2.7 ND Specification

We consider two types of properties that ND protocols should satisfy. The first one pertains to *correctness*, expressed through property ND1 (Fig. 4): If two correct nodes[2] are declared neighbors at some time, then they must indeed be neighbors at that time. More precisely, there are two cases: 1) Node A can declare that B is its neighbor (i.e., A can receive messages from B) or 2) A can declare that it is a neighbor of C (i.e., C can receive messages from A). In the latter case, property ND1 requires link (C,A) to be up at not exactly time t', but rather $(dist(A,C) + nlos(A,C))\mathbf{v}^{-1}$ (propagation delay) after t'. As our model mandates that the link state is determined at the receiving end (node), if A declares that it is a neighbor of C at time t', a message sent by A at t would be indeed received by C. In other words, A is not forced to estimate the propagation delay to make a correct neighbor statement.

The second type of property pertains to *availability*, expressed through property ND2 (Fig. 4), tailored to T-protocols. An additional notion needs to be introduced to formulate satisfiable availability properties: *ND range*, $\mathbf{R} \in \mathbb{R}_{>0}$. Typically, $\mathbf{R}$ is equal to the *nominal communication range* for a given wireless medium and transceiver technology. However, we use $\mathbf{R}$ more freely as the communication range[3] for which ND inferences are drawn. In other words, nodes at a communication range larger than $\mathbf{R}$ will not be required to declare each other neighbors.

2. The requirement that B and C be correct is explained in Section 5.

3. By "communication range," we understand the actual distance plus NLOS effects.

$$\text{ND1} \quad \forall \mathcal{S} \in \mathbb{S}, \theta \in \Theta_{\mathcal{S},\mathcal{P},\mathcal{A}}.\ \forall A, B, C \in V_{\text{cor}}, t, t' \in \mathbb{R}_{\geq 0}.\ \ \mathsf{Neighbor}(A; t; B, C, t') \in \theta \implies (C = A \ \wedge\ link(B \rightarrow A, t')) \ \vee\ (B = A \ \wedge\ link(A \rightarrow C, t' + (dist(A, C) + nlos(A, C))\mathbf{v}^{-1}))$$

$$\text{ND2} \quad \forall d \in (0, \mathbf{R}].\ \forall A, B \in \mathbb{V}, A \neq B.\ \exists \mathcal{S} \in \mathbb{S}.\ V = V_{\mathbf{cor}} = \{A, B\} \ \wedge\ dist(A, B) = d \ \wedge\ link(A \leftrightarrow B, \mathbb{R}_{\geq 0}) \ \wedge\ \exists \theta \in \Theta_{\mathcal{S},\mathcal{P},\mathcal{A}}.\ \mathsf{Neighbor}(A; t; B, A, t') \in \theta$$

Fig. 4. Basic ND properties.

Property ND2 requires that for every distance d in the desired ND range $\mathbf{R}$, there should be at least some setting in which the protocol is able to conclude that a node is a neighbor (in some, not all executions); this setting should contain exactly two nodes, both *correct* and at distance d, being neighbors. The "two-nodes setting" requirement clarifies why we call this *two-party* ND. The ND2 property is the least that can be required from a usable two-party ND protocol: Indeed, a protocol not satisfying this property would be unable to conclude, for some distance(s) in the ND range, that nodes are neighbors. This makes the impossibility result in Section 3 more meaningful: impossibility with respect to a weak property implies impossibility for any stronger property.

3 Impossibility for T-Protocols

We show in this section that no time-based protocol can solve the two-party ND problem as specified by properties ND1 and ND2 in Fig. 4. We base the proof on the fact that it is impossible for a correct node to distinguish between different settings based on a T-local view. This is captured by Lemma 1 (for the proof, please see [26]). Our impossibility result, Theorem 1 below, stems from showing two settings that are indistinguishable by a correct node, one with two nodes being neighbors and one where they are not (Fig. 5). We elaborate on the assumptions and implications of this result in Section 5.

We emphasize that the nonrestricted form of the message space $\mathbb{M}$ encompasses all possible messages including, for example, time stamps and any type of cryptography, thus contributing to the generality of the impossibility result.

Lemma 1. *Let $\mathcal{P}$ be a T-protocol model, $\mathcal{S}$ and $\mathcal{S}'$ be settings such that $V_{\text{cor}} = V'_{\text{cor}}$, and $\theta \in \Theta_{\mathcal{S},\mathcal{P}}$ and $\theta' \in \Theta_{\mathcal{S}'}$ be traces such that local traces $\theta|_A = \theta'|_A$ for all $A \in V_{\text{cor}}$. Then θ' is feasible with respect to T-protocol model $\mathcal{P}$.*

Theorem 1. *If $\Delta_{\text{relay}} \leq \frac{\mathbf{R}}{\mathbf{v}}$, then there exists no T-protocol model which satisfies ND1 and ND2 (Fig. 4) for the adversary model $\mathcal{A}_{\Delta_{\text{relay}}}$.*

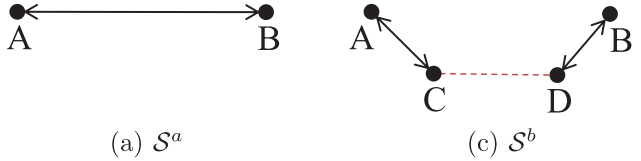

Fig. 5. Settings used in the impossibility result proof. Settings $\mathcal{S}^a = \langle \{A, B\}, loc^a, type^a, link^a, nlos^a \rangle$ and $\mathcal{S}^b = \langle \{A, B, C\}, loc^b, type^b, link^b, nlos^b \rangle$. In both settings, nodes A and B are correct, nodes C and D are adversarial. The location functions are such that $dist^b(A, C) + dist^b(D, B) + \mathbf{v}\mathbf{v}_{\text{adv}}{}^{-1} dist^b(C, D) + \mathbf{v}\Delta_{\text{relay}} \leq dist^a(A, B)$. The state of links does not change over time and is shown in the figure (lack of arrow means that the link is down). For all links and settings, $nlos = 0$. The dashed line in (b) denotes the adversarial channel.

Proof. To prove that under the assumptions of the theorem no T-protocol model can satisfy both ND1 and ND2, we show that any T-protocol model that satisfies ND2 cannot satisfy ND1.

Take any T-protocol model $\mathcal{P}$ satisfying ND2. Pick some distance $d \geq \mathbf{v}\Delta_{\text{relay}}$ in the ND range ($d \leq \mathbf{R}$). Property ND2 guarantees the existence of a setting such as the one shown in Fig. 5a (we denote it $\mathcal{S}^a$) and the existence of a trace $\theta \in \Theta_{\mathcal{S}^a,\mathcal{P},\mathcal{A}_{\Delta_{\text{relay}}}}$ such that $\mathsf{Neighbor}(A; t; B, t') \in \theta$. As θ is feasible with respect to setting $\mathcal{S}^a$, this trace has to be of the form shown in Fig. 6.

In setting $\mathcal{S}^b$, shown in Fig. 5b, we have $link(B \not\rightarrow A, \mathbb{R}_{\geq 0})$. Consider the trace θ' (Fig. 6), which is essentially the same as θ, but for nodes C and D relaying all the communication between nodes A and B. It is simple to check that this trace is feasible with respect to setting $\mathcal{S}^b$. It is also feasible with respect to T-protocol model $\mathcal{P}$: This follows from Lemma 1, as $\theta|_{A,t} = \theta'|_{A,t}$ and $\theta|_{B,t} = \theta'|_{B,t}$. Finally, θ' is feasible with respect to the adversary model $\mathcal{A}_{\Delta_{\text{relay}}}$, because $\delta_2 - \delta_1 = \delta_4 - \delta_3 \geq \Delta_{\text{relay}} + dist(C, D)\mathbf{v}_{\text{adv}}^{-1}$. Therefore, θ' belongs to $\Theta_{\mathcal{S}^b,\mathcal{P},\mathcal{A}_{\Delta_{\text{relay}}}}$ and together with $\mathcal{S}^b$ forms the counterexample that we were looking for: A concludes B is a neighbor whereas it is not. Thus, T-protocol model $\mathcal{P}$ does not satisfy ND1. As $\mathcal{P}$ was chosen arbitrarily, this concludes the proof. □

4 ND Protocols

We consider here four types of ND protocols, with one representative protocol per type. We distinguish between 1) beacon-based protocols (*B-protocols*), represented by $\mathcal{P}^{\text{B/T}}$ and $\mathcal{P}^{\text{B/TL}}$, which require the transmission of one message by one of the protocol participants and synchronized clocks, and 2) challenge-response protocols (*CR-protocols*), represented by $\mathcal{P}^{\text{CR/T}}$ and $\mathcal{P}^{\text{CR/TL}}$, which require a transmission of messages by both participants but *no* synchronized clocks. Within and across these categories, we distinguish protocols according to their capability to perform time measurements (T-protocols) or time measurements and location awareness (TL-protocols).

Fundamentally, beyond authentication mechanisms, all the ND protocols we consider measure the signal time-of-flight (ToF) between two nodes: B-protocols, with tightly synchronized clocks, are able to estimate ToF by transmitting a single beacon message, whereas CR-protocols require two messages, a challenge and a response, for the same purpose. T-protocols accept neighbor relations as valid if the ToF distance is below a threshold, whereas TL-protocols require this distance to be equal to the geographical distance calculated based on nodes locations. We first introduced such TL-protocols in [27], [28]. In contrast, T-protocols come from earlier related work [15], [9].

$$\begin{aligned}\theta = &\{\mathsf{Bcast}(A;t_i;m_i) \mid i \in I_A\} \cup \{\mathsf{Receive}(B;t_i+\Delta;m_i) \mid i \in I_A\} \cup\\ &\{\mathsf{Bcast}(B;t_i;m_i) \mid i \in I_B\} \cup \{\mathsf{Receive}(A;t_i+\Delta;m_i) \mid i \in I_B\} \cup\\ &\{\mathsf{Neighbor}(A;t_i;A,B,t_i') \mid i \in J_A^A\} \cup \{\mathsf{Neighbor}(A;t_i;B,A,t_i') \mid i \in J_A^B\} \cup\\ &\{\mathsf{Neighbor}(B;t_i;A,B,t_i') \mid i \in J_B^A\} \cup \{\mathsf{Neighbor}(B;t_i;B,A,t_i') \mid i \in J_B^B\}\end{aligned}$$

$$\begin{aligned}\theta' = &\{\mathsf{Bcast}(A;t_i;m_i) \mid i \in I_A\} \cup \{\mathsf{Receive}(C;t_i+\delta_1;m_i) \mid i \in I_A\} \cup\\ &\{\mathsf{Bcast}(D;t_i+\delta_2;m_i) \mid i \in I_A\} \cup \{\mathsf{Receive}(B;t_i+\Delta;m_i) \mid i \in I_A\} \cup\\ &\{\mathsf{Bcast}(B;t_i;m_i) \mid i \in I_B\} \cup \{\mathsf{Receive}(D;t_i+\delta_3;m_i) \mid i \in I_B\} \cup\\ &\{\mathsf{Bcast}(C;t_i+\delta_4;m_i) \mid i \in I_B\} \cup \{\mathsf{Receive}(A;t_i+\Delta;B,m_i) \mid i \in I_B\} \cup\\ &\{\mathsf{Neighbor}(A;t_i;A,B,t_i') \mid i \in J_A^A\} \cup \{\mathsf{Neighbor}(A;t_i;B,A,t_i') \mid i \in J_A^B\} \cup\\ &\{\mathsf{Neighbor}(B;t_i;A,B,t_i') \mid i \in J_B^A\} \cup \{\mathsf{Neighbor}(B;t_i;B,A,t_i') \mid i \in J_B^B\}\end{aligned}$$

Fig. 6. Traces used in the proof of Theorem 1; $\Delta = dist^a(A,B)\mathbf{v}^{-1}$, $t_i, t_i' \in \mathbb{R}_{\geq 0}$, and $I_A, I_B, J_A^A, J_A^B, J_B^A, J_B^B$ are pairwise disjoint index sets with $J_A^B \neq \emptyset$ (all the other index sets can be empty); $\delta_1 = dist^b(A,C)\mathbf{v}^{-1}$, $\delta_2 = \Delta - dist^b(D,B)\mathbf{v}^{-1}$, $\delta_3 = dist^b(B,D)\mathbf{v}^{-1}\delta_4 = \Delta - dist^b(C,A)\mathbf{v}^{-1}$.

F1 $\forall A,B \in V_{\text{cor}}, t_1,t_2 \in \mathbb{R}_{\geq 0}, n \in \mathsf{Nonces}, m_1 \in \mathbb{M}.\ \ n \sqsubseteq m_1 \ \wedge\ \mathsf{Fresh}(A;t_1;n) \in \theta$
$\wedge\ \mathsf{Bcast}(B;t_2;n \sqsubseteq m_1) \in \theta \implies (A = B \ \wedge\ t_2 \geq t_1) \ \vee\ (A \neq B \ \wedge\ \exists \delta \geq \Delta_{\text{relay}}, m_2 \in \mathbb{M}.$
$n \sqsubseteq m_2 \ \wedge\ \mathsf{Receive}(B;t_2-\delta;n \sqsubseteq m_2) \in \theta)$

A1 $\forall A \in V_{\text{cor}}, B \in V_{\text{adv}}, t_1,t_2 \in \mathbb{R}_{\geq 0}, n \in \mathsf{Nonces}, m_1 \in \mathbb{M}.\ \ n \sqsubseteq m_1 \ \wedge\ \mathsf{Fresh}(A;t_1;n) \in \theta$
$\wedge\ \mathsf{Bcast}(B;t_2;n \sqsubseteq m_1) \in \theta \implies \exists C \in V_{\text{adv}}, \delta \geq \Delta_{\text{relay}} + dist(C,B)\mathbf{v}_{\text{adv}}^{-1}, m_2 \in \mathbb{M}.\ \ n \sqsubseteq m_2$
$\wedge\ \mathsf{Receive}(C;t_2-\delta;n \sqsubseteq m_2) \in \theta$

A2 $\forall A \in V_{\text{adv}}, B \in \mathbb{V}, t \in \mathbb{R}_{\geq 0}, m, m_0, m_1 \in \mathbb{M}.\ \ m = \mathsf{auth}_B(m_0) \sqsubseteq m_1$
$\wedge\ \mathsf{Bcast}(A;t;m \sqsubseteq m_1) \in \theta \implies (B \in V_{\text{adv}})$
$\vee\ (\exists C \in V_{\text{adv}}, \delta \geq \Delta_{\text{relay}} + dist(C,A)\mathbf{v}_{\text{adv}}^{-1}, m_2 \in \mathbb{M}.\ \ m \sqsubseteq m_2 \ \wedge\ \mathsf{Receive}(C;t-\delta;m \sqsubseteq m_2) \in \theta)$

A3 $\forall A \in V_{\text{adv}}, B,C \in \mathbb{V}, t \in \mathbb{R}_{\geq 0}, m, m_0, m_1 \in \mathbb{M}.\ \ m = \mathsf{auth}_{BC}(m_0) \sqsubseteq m_1$
$\wedge\ \mathsf{Bcast}(A;t;m \sqsubseteq m_1) \in \theta \implies (B \in V_{\text{adv}}) \ \vee\ (C \in V_{\text{adv}})$
$\vee\ (\exists D \in V_{\text{adv}}, \delta \geq \Delta_{\text{relay}} + dist(D,A)\mathbf{v}_{\text{adv}}^{-1}, m_2 \in \mathbb{M}.\ \ m \sqsubseteq m_2 \ \wedge\ \mathsf{Receive}(D;t-\delta;m \sqsubseteq m_2) \in \theta)$

Fig. 7. Adversary- and common protocol-feasibility rules.

4.1 Message Space

We define next the message space $\mathbb{M}$ (unlike the unspecified one for the impossibility result). Any of the following is a message:

- an identifier $A \in \mathbb{W}$,
- a timestamp $t \in \mathbb{R}_{\geq 0}$,
- a location $l \in \mathbb{R}^3$, or
- a nonce $n \in \mathsf{Nonces}$.

Moreover, two messages m_1, m_2 can be *concatenated* to form a message $\langle m_1, m_2 \rangle$. Furthermore, an *asymmetric authenticator* $\text{auth}_A(m)$ and a *symmetric authenticator* $\text{auth}_{AB}(m)$, where $A, B \in \mathbb{W}$ and $m \in \mathbb{M}$, are also messages.[4] For symmetric authenticators we assume that $\text{auth}_{AB}(m) = \text{auth}_{BA}(m)$. Essentially, messages are terms, with the subterm relation is denoted by $\sqsubseteq$.

Every message m has a *duration* $|m| \in \mathbb{R}_{\geq 0}$, which determines the transmission delay (*not* including the propagation delay), reflecting the bit-rate of the underlying communication technology. We assume that message duration is preserved by concatenation, but not by an authenticator. For $m = \langle m_1, m_2, \ldots, m_k \rangle$, the duration is $|m| = |m_1| + |m_2| + \cdots + |m_k|$ and the *position* of m_i in m is $pos(m_i \sqsubseteq m) = |m_1| + \cdots + |m_{i-1}|$, with $pos(m_1 \sqsubseteq m) = 0$; in the case of multiple occurrences of $m' \sqsubseteq m$, $pos(m' \sqsubseteq m)$ gives the position of the first occurrence. When we use the duration function for any concatenated message, we omit the brackets: $|m_1, m_2, \ldots, m_k|$. Finally, we assume that the duration of identifiers, time stamps, locations, nonces, and authenticators in $\mathbb{M}$ is upper bounded by some constant.

Notation: Assuming that $m_1 \sqsubseteq m_2$, we use

$$\text{Bcast}(A;t;m_1 \sqsubseteq m_2)$$

to denote the event

$$\text{Bcast}(A;t - pos(m_1 \sqsubseteq m_2);m_2).$$

Likewise for `Receive`.

4.2 Protocol-Feasible Traces

In Section 2.5, we have defined feasibility rules for arbitrary protocols. However, for reasoning about specific protocols, it is more convenient to define protocols with rules, rather than specifying a protocol model function and applying the general rules in Fig. 2.

The rules that specify this type of feasibility are protocol-dependent and are defined in Section 4.5. However, there is one general rule that dictates the behavior of correct nodes with respect to nonces. Rule F1 (Fig. 7) guarantees that if a nonce n is freshly generated at time t then 1) the node that generated n will not broadcast it *before* t, and 2) any other correct node who broadcasts a message containing nonce n must have received it (possibly in a different message) at least Δ_{relay} before broadcasting; this time difference is measured with respect to the positions of the nonce in the respective messages.

4. Examples of asymmetric authenticators are digital signatures; examples of symmetric authenticators are message authentication codes.

$$\mathsf{ND2}^{\mathrm{B/T}} \quad \forall \mathcal{S} \in \mathbb{S}, \theta \in \Theta_{\mathcal{S},\mathcal{P},\mathcal{A}}.\ \forall A, B \in \mathbb{V}_{\mathrm{cor}}, t \in \mathbb{R}_{\geq 0}.\ \mathsf{NDstart}(A;t) \in \theta \ \wedge\ link(A{\rightarrow}B,[t,t+T_{\mathcal{P}}]) \wedge dist(A,B)+nlos(A,B) \leq \mathbf{R} \implies \exists t' \in [t,\infty), t'' \in [t,t+T_{\mathcal{P}}].\ \mathsf{Neighbor}(B;t';A,B,t'') \in \theta$$

$$\mathsf{ND2}^{\mathrm{B/TL}} \quad \forall \mathcal{S} \in \mathbb{S}, \theta \in \Theta_{\mathcal{S},\mathcal{P},\mathcal{A}}.\ \forall A, B \in \mathbb{V}_{\mathrm{cor}}, t \in \mathbb{R}_{\geq 0}.\ \mathsf{NDstart}(A;t;B) \in \theta \ \wedge\ link(A{\rightarrow}B,[t,t+T_{\mathcal{P}}]) \wedge nlos(A,B)=0 \implies \exists t' \in [t,\infty), t'' \in [t,t+T_{\mathcal{P}}].\ \mathsf{Neighbor}(B;t';A,B,t'') \in \theta$$

$$\mathsf{ND2}^{\mathrm{CR/T}} \quad \forall \mathcal{S} \in \mathbb{S}, \theta \in \Theta_{\mathcal{S},\mathcal{P},\mathcal{A}}.\ \forall A, B \in \mathbb{V}_{\mathrm{cor}}, t \in \mathbb{R}_{\geq 0}.\ \mathsf{NDstart}(A;t;B) \in \theta \ \wedge\ link(A{\leftrightarrow}B,[t,t+T_{\mathcal{P}}]) \wedge dist(A,B)+nlos(A,B) \leq \mathbf{R} \implies \exists t_1, t_2 \in [t,\infty), t', t'' \in [t,t+T_{\mathcal{P}}].\ \mathsf{Neighbor}(A;t_1;A,B,t') \in \theta \ \wedge\ \mathsf{Neighbor}(A;t_2;B,A,t'') \in \theta$$

$$\mathsf{ND2}^{\mathrm{CR/TL}} \quad \forall \mathcal{S} \in \mathbb{S}, \theta \in \Theta_{\mathcal{S},\mathcal{P},\mathcal{A}}.\ \forall A, B \in \mathbb{V}_{\mathrm{cor}}, t \in \mathbb{R}_{\geq 0}.\ \mathsf{NDstart}(A;t;B) \in \theta \ \wedge\ link(A{\leftrightarrow}B,[t,t+T_{\mathcal{P}}]) \wedge nlos(A,B)=0 \implies \exists t_1, t_2 \in [t,\infty), t', t'' \in [t,t+T_{\mathcal{P}}].\ \mathsf{Neighbor}(A;t_1;A,B,t') \in \theta \wedge \mathsf{Neighbor}(A;t_2;B,A,t'') \in \theta$$

Fig. 8. ND availability properties.

4.3 Adversary-Feasible Traces

To reason about the security of specific protocols, we consider an adversary model, $\mathcal{A}^{\mathcal{P}}_{\Delta_{\mathrm{relay}}}$, stronger than that defined in Section 2.6. Intuitively, adversarial nodes are allowed to send arbitrary messages, except for messages that would violate properties of authenticators or freshness; these have to be relayed with the relaying delay at least Δ_{relay}.

A trace θ is feasible with respect to $\mathcal{A}^{\mathcal{P}}_{\Delta_{\mathrm{relay}}}$ if rules A1-A3 (Fig. 7) are satisfied. Rules A2 and A3 deal with authenticators: An adversarial node is allowed to send a message containing arbitrary authenticators, as long as these authenticators can be generated by itself or another adversarial node. This implies that adversarial nodes can share cryptographic keys or any material used for authentication. Furthermore, rules A2 and A3 reflect that the adversary cannot forge authenticated messages: Any message sent by an adversarial node that contains a correct node authenticator must be relayed. In other words, some (possibly the same) adversarial node must have received a message containing this authenticator earlier, at least Δ_{relay} plus the propagation delay between the two nodes over the adversarial channel. This condition reflects the structure of the adversarial channel: Any two adversarial nodes can establish direct communication. Rule A1 is similar to A2, but it is responsible for freshness: An adversary sending a message with a nonce generated by a correct node can only be relaying the message (nonce). In this sense, rule A1 is an adversarial equivalent of rule F1.

4.4 ND Specification

To reason about the security of specific protocols, we use the correctness property ND1 as introduced in Fig. 4, but we provide stronger *protocol-type specific* availability properties. Informally, we require that if nodes are neighbors for a long enough protocol-specific time $T_{\mathcal{P}}$, the protocol must declare them neighbors.

Fig. 8 displays ND2 properties for all types of protocols we consider. These properties differ in four aspects, one depending on whether the protocol is T or TL, whereas the other three aspects depending on the protocol is beacon or challenge response. The first aspect is the NDstart event: For CR-protocols, a particular neighbor B with which ND is started is specified, whereas no such specification is necessary for B-protocols. Second, it may be required that link (A,B) be up only in one direction (B-protocols) or both directions (CR-protocols). Third, for T-protocols, an upper-bound on propagation distance in enforced ($dist(A,B)+\ nlos(A,B) \leq \mathbf{R}$), whereas for TL-protocols line-of-sight propagation is required ($nlos(A,B)=0$). Fourth, different forms of neighbor declaration are possible. The node making the declaration might be the same as (CR-protocols) or different (B-protocols) from the one initiating the ND protocol. Moreover the declaration might be unidirectional (B-protocols) or bidirectional (CR-protocols).

4.5 Protocol Definitions

The protocols are formally defined by rules as the ones in Fig. 10. To make the presentation more approachable, we first present the protocols informally in the form of pseudocode, and then we describe how the rules model the behavior of the protocol. The pseudocode is divided into *blocks*, each starting with a *triggering* event (*on* clause). Upon a triggering event, the block body is executed, i.e., other events take place. Defining protocol rules from the pseudocode is for the most part straightforward, and could be (partially) automated. The only nontrivial part is related to composability of the protocol. We explain this process next.

We start with a simple B/T-protocol we denote $\mathcal{P}^{\mathrm{B/T}}$, which is essentially the *temporal packet leash* protocol proposed by Hu et al. [15]. The pseudocode is shown in Fig. 9, the rules defining the protocol are presented in Fig. 10. Blocks 1-2 describe the behavior after the ND protocol is started at node A (e.g., by a higher layer protocol); P1 and P2 are the two rules that correspond to this block. Blocks 3-5 describe the behavior of a node after it receives a beacon message, and it is modeled by rules P3 and P4. Rule P1 is straightforward: It ensures that if the triggering event of blocks 1-2, $\mathtt{NDstart}(A;t_1)$, occurs in the trace, the event in the body of the block also occurs. In the same fashion, rule P3 is defined for blocks 3-5, with an additional condition coming from the *if* clause.

These two rules are already sufficient to prove the ND2 property, but they only define half of aspects of the protocol

```
1:  on NDstart(A; t1)
2:    Bcast(A; t1; ⟨A, t1, auth_A(t1)⟩)
3:  on Receive(B; t2; ⟨A, t1, auth_A(t1)⟩)
4:    if t2 − t1 ≤ Rv^−1
5:      Neighbor(B; t2 + |A, t1, auth_A(t1)|; A, B, t2)
```

Fig. 9. Pseudocode for protocol $\mathcal{P}^{\mathrm{B/T}}$.

$$\begin{aligned}
&\text{P1} \quad \forall A \in V_{\text{cor}}, t_1 \in \mathbb{R}_{\geq 0}.\ \ \mathsf{NDstart}(A; t_1) \in \theta \implies \mathsf{Bcast}(A; t_1; \langle A, t_1, \mathsf{auth}_A(t_1)\rangle) \in \theta \\
&\text{P2} \quad \forall A \in V_{\text{cor}}, B \in \mathbb{V}, t_1, t \in \mathbb{R}_{\geq 0}, m \in \mathbb{M}.\ \ \mathsf{auth}_B(t) \sqsubseteq m \ \wedge\ \mathsf{Bcast}(A; t_1; m) \in \theta \\
&\qquad \implies m = \langle A, t_1, \mathsf{auth}_A(t_1)\rangle \\
&\text{P3} \quad \forall B \in V_{\text{cor}}, A \in \mathbb{V}, t_1, t_2 \in \mathbb{R}_{\geq 0}.\ \ \mathsf{Receive}(B; t_2; \langle A, t_1, \mathsf{auth}_A(t_1)\rangle) \in \theta \ \wedge\ t_2 - t_1 \leq \mathbf{Rv}^{-1} \\
&\qquad \implies \mathsf{Neighbor}(B; t_2 + |A, t_1, \mathsf{auth}_A(t_1)|; A, B, t_2) \in \theta \\
&\text{P4} \quad \forall B \in V_{\text{cor}}, A, C \in \mathbb{V}, t_2, t \in \mathbb{R}_{\geq 0}.\ \ \mathsf{Neighbor}(B; t; A, C, t_2) \in \theta \implies C = B \\
&\qquad \wedge\ \exists t_1 \in \mathbb{R}_{\geq 0}.\ \ \mathsf{Receive}(B; t_2; \langle A, t_1, \mathsf{auth}_A(t_1)\rangle) \in \theta \ \wedge\ t_2 - t_1 \leq \mathbf{Rv}^{-1} \\
&\qquad \wedge\ t = t_2 + |A, t_1, \mathsf{auth}_A(t_1)|
\end{aligned}$$

Fig. 10. Rules defining protocol $\mathcal{P}^{\text{B/T}}$.

functionality. Indeed, nothing prevents a node running this protocol from making arbitrary neighbor declarations. Rule P4 addresses this, stating that if a node makes a neighbor declaration, this has to be done according to blocks 3-5, i.e., the node had to receive a "fresh enough" beacon message. Only one aspect remains: Correct nodes are still allowed to broadcast arbitrary messages, including bogus beacon messages. This is addressed by rule P2. To motivate the definition of P2, let us consider an alternative rule would still be coherent with the pseudocode: If a correct node broadcasts a message at time t_1, this message is $\langle A, t_1, \text{auth}_A(t_1)\rangle$. We can prove that such a protocol satisfies the ND specification. However, this would be a weak result, precisely because that rule states that correct nodes cannot send any other messages than beacons. If the ND protocol were used along with or by any other protocol, obviously using other forms of messages, the result would no longer apply. To circumvent this undesired composability restriction, rule P2 only requires that if a correct node broadcasts at t_1 a message m of a *particular form*, i.e., containing $\text{auth}_B(t)$ as a subterm, then $m = \langle A, t_1, \mathsf{auth}_A(t_1)\rangle$. Hence, rule P2 gives a much less restrictive condition on protocols that can be securely composed with $\mathcal{P}^{\text{B/T}}$: basically, it mandates that any other protocol does not use authenticated timestamps of this form.[5] Rule P4, in terms of composability, implies that the node cannot run any other ND protocol (i.e., a protocol making neighbor declarations), but we do not see this as a real restriction.

Next, we describe $\mathcal{P}^{\text{CR/TL}}$, a CR/TL-protocol (pseudocode Fig. 11 rules Fig. 12). This protocol has a practical design twist: As authentication of a message can be a time-consuming process, in this protocol, we remove it from the time-critical ToF estimation phase. Otherwise, if the response needs too much time to be calculated, the clock of the challenging node can drift beyond an acceptable accuracy level. A protocol parameter $\Delta \in \mathbb{R}_{\geq 0}$ determines exactly how long after the challenge reception a node replies.

We assume that a node keeps track of all the events it observes, and it can always refer to this "history," as in 10-12. There is no explicit block responsible for receiving the $\langle n_2\rangle$ response sent by B in 06; node A does not take any action other than recording the event occurrence, for later reference in line 11.

Considering again that "triggering event implies block body events," rule P1 is defined for blocks 01-03, P2 for blocks 04-08, and P4 for blocks 09-15. We do not define rules that restrict the occurrence of Fresh events (in lines 02 and 05) or the form of broadcasted messages (in lines 03 and 06), so that there is no obstacle for composability. For line 08, rule P3 is defined: If a node broadcasts a message m containing an authenticator of the form $\text{auth}_B(n_1, n_2, l)$, then m is precisely the message defined in line 08, and all the other events from block 04-08 occur. Finally, rule P5 is defined based on blocks 09-15. There is only one rule, despite two Neighbor events in lines 14 and 15, because both events match the universally quantified Neighbor event in P5; The rule uses a disjunction, as there are (small) timing differences in the node behavior depending on which of these two event is considered.

The pesudocode for protocols $\mathcal{P}^{\text{B/TL}}$ and $\mathcal{P}^{\text{CR/T}}$ is shown in Figs. 13 and 14. The rules defining these protocols are available in [26]. They are omitted here as they are similar to those for the $\mathcal{P}^{\text{B/T}}$ and $\mathcal{P}^{\text{CR/TL}}$ protocols. We note, however, that opposite to the other protocols, $\mathcal{P}^{\text{B/TL}}$ relies on symmetric authenticators. The purpose of this is to demonstrate that the protocols can be modified to work with symmetric cryptography. There is no specific reason why we chose $\mathcal{P}^{\text{B/TL}}$ for this demonstration.

4.6 Results and Isabelle/HOL Mechanization

We prove that the protocols presented in Section 4.5 provide secure ND, which is summarized by the following theorem:

Theorem 2. *The protocols $\mathcal{P}^{\text{B/T}}$, $\mathcal{P}^{\text{CR/TL}}$, $\mathcal{P}^{\text{CR/T}}$, and $\mathcal{P}^{\text{B/TL}}$ satisfy* ND1 *and* ND2 *(the appropriate variant, Fig. 8) for the adversary model $\mathcal{A}^{\mathcal{P}}_{\Delta_{\text{relay}}}$ under the assumptions summarized in Table 1.*

```
01:  on NDstart(A; t1; B)
02:    Fresh(A; t1 + |B|; n1)
03:    Bcast(A; t1; ⟨B, n1⟩)
04:  on Receive(B; t; ⟨B, n1⟩)
05:    Fresh(B; t + Δ; n2)
06:    Bcast(B; t + Δ; ⟨n2⟩)
07:    let τ > Δ
08:    Bcast(B; t + τ; ⟨loc(B), auth_B(n1, n2, loc(B))⟩)
09:  on Receive(A; t; ⟨l, auth_B(n1, n2, l)⟩)
10:    if occurred Fresh(A; t1 + |B|; n1)
11:    if occurred Bcast(A; t1; ⟨B, n1⟩)
12:    if occurred Receive(A; t2; ⟨n2⟩)
13:    if v(t2 − t1 − Δ) = 2d(loc(A), l)
14:      Neighbor(A; t + |l, auth_B(n1, n2, l)|; A, B, t1)
15:      Neighbor(A; t + |l, auth_B(n1, n2, l)|; B, A, t2)
```

Fig. 11. Pseudocode for protocol $\mathcal{P}^{\text{CR/TL}}$.

5. If this would pose a problem, the protocol can be modified, for example, by authenticating a time stamp concatenated with some constant rather than just a time stamp.

$$
\begin{aligned}
&\text{P1}\quad \forall A \in V_{\text{cor}}, B \in \mathbb{V}, t_1 \in \mathbb{R}_{\geq 0}.\ \ \mathsf{NDstart}(A; t_1; B) \in \theta \implies \exists n_1 \in \mathsf{Nonces}.\\
&\quad \mathsf{Fresh}(A; t_1 + |B|; n_1) \in \theta \ \wedge\ \mathsf{Bcast}(A; t_1; \langle B, n_1 \rangle) \in \theta\\
&\text{P2}\quad \forall B \in V_{\text{cor}}, t \in \mathbb{R}_{\geq 0}, n_1 \in \mathsf{Nonces}.\ \ \mathsf{Receive}(B; t; \langle B, n_1 \rangle) \in \theta \implies \exists n_2 \in \mathsf{Nonces}, \tau > \Delta.\\
&\quad \mathsf{Fresh}(B; t + \Delta; n_2) \in \theta \ \wedge\ \mathsf{Bcast}(B; t + \Delta; \langle n_2 \rangle) \in \theta\\
&\quad \wedge\ \mathsf{Bcast}(B; t + \tau; \langle loc(B), \mathsf{auth}_B(n_1, n_2, loc(B)) \rangle) \in \theta\\
&\text{P3}\quad \forall B \in V_{\text{cor}}, C \in \mathbb{V}, t \in \mathbb{R}_{\geq 0}, n_1, n_2 \in \mathsf{Nonces}, l \in \mathbb{R}^3, m \in \mathbb{M}.\ \ \mathsf{auth}_C(n_1, n_2, l) \sqsubseteq m\\
&\quad \wedge\ \mathsf{Bcast}(B; t; m) \in \theta \implies \exists \tau > 0.\ \ m = \langle loc(B), \mathsf{auth}_B(n_1, n_2, loc(B)) \rangle\\
&\quad \wedge\ \mathsf{Receive}(B; t - \tau - \Delta; \langle B, n_1 \rangle) \in \theta \ \wedge\ \mathsf{Fresh}(B; t - \tau; n_2) \in \theta \ \wedge\ \mathsf{Bcast}(B; t - \tau; \langle n_2 \rangle) \in \theta\\
&\text{P4}\quad \forall A \in V_{\text{cor}}, B \in \mathbb{V}, n_1, n_2 \in \mathsf{Nonces}, t_1, t_2, t \in \mathbb{R}_{\geq 0}, l \in \mathbb{R}^3.\\
&\quad \mathsf{Receive}(A; t; \langle l, \mathsf{auth}_B(n_1, n_2, l) \rangle) \in \theta \ \wedge\ \mathsf{Fresh}(A; t_1 + |B|; n_1) \in \theta\\
&\quad \wedge\ \mathsf{Bcast}(A; t_1; \langle B, n_1 \rangle) \in \theta \ \wedge\ \mathsf{Receive}(A; t_2; \langle n_2 \rangle) \in \theta \ \wedge\ \mathbf{v}(t_2 - t_1 - \Delta) = 2d(loc(A), l) \implies\\
&\quad \mathsf{Neighbor}(A; t + |l, \mathsf{auth}_B(n_1, n_2, l)|; A, B, t_1) \in \theta \ \wedge\ \mathsf{Neighbor}(A; t + |l, \mathsf{auth}_B(n_1, n_2, l)|; B, A, t_2) \in \theta\\
&\text{P5}\quad \forall A \in V_{\text{cor}}, B, C \in \mathbb{V}, t, t_0 \in \mathbb{R}_{\geq 0}.\ \ \mathsf{Neighbor}(A; t; B, C, t_0) \in \theta \implies\\
&\quad (C = A \ \wedge\ \exists n_1, n_2 \in \mathsf{Nonces}, t_1 \in \mathbb{R}_{\geq 0}, l \in \mathbb{R}^3.\ \ \mathsf{Fresh}(A; t_1 + |B|; n_1) \in \theta\\
&\quad \wedge\ \mathsf{Bcast}(A; t_1; \langle B, n_1 \rangle) \in \theta \ \wedge\ \mathsf{Receive}(A; t_0; \langle n_2 \rangle) \in \theta\\
&\quad \wedge\ \mathsf{Receive}(A; t - |l, \mathsf{auth}_B(n_1, n_2, l)|; \langle l, \mathsf{auth}_B(n_1, n_2, l) \rangle) \in \theta \ \wedge\ \mathbf{v}(t_0 - t_1 - \Delta) = 2d(loc(A), l))\\
&\quad \vee\\
&\quad (B = A \ \wedge\ \exists n_1, n_2 \in \mathsf{Nonces}, t_2 \in \mathbb{R}_{\geq 0}, l \in \mathbb{R}^3.\ \ \mathsf{Fresh}(A; t_0 + |C|; n_1) \in \theta\\
&\quad \wedge\ \mathsf{Bcast}(A; t_0; \langle C, n_1 \rangle) \in \theta \ \wedge\ \mathsf{Receive}(A; t_2; \langle n_2 \rangle) \in \theta\\
&\quad \wedge\ \mathsf{Receive}(A; t - |l, \mathsf{auth}_C(n_1, n_2, l)|; \langle l, \mathsf{auth}_C(n_1, n_2, l) \rangle) \in \theta \ \wedge\ \mathbf{v}(t_2 - t_0 - \Delta) = 2d(loc(A), l))
\end{aligned}
$$

Fig. 12. Rules defining protocol $\mathcal{P}^{\text{CR/TL}}$.

We comment further on these results in Section 5.2. We formalize our framework, with a few minor modifications, in the theorem prover Isabelle [19] with higher-order logic (HOL). This allows us to mechanically verify the proofs, greatly increasing the confidence in the results. The pen-and-paper proofs for Theorem 2 and the source code for the mechanized proofs are available in [26]. We use an extension of HOL, the HOL-Complex logic that defines complex and real numbers, because our model requires the latter.

In the Isabelle formalization process, we make one noteworthy modification compared to our pen-and-paper proofs: we model concatenated messages as lists of simple messages (identifiers, time stamps, locations, nonces, authenticators). With this representation, we have a one-to-one mapping between the messages in the model, and "real-world" messages. Whereas one "real-world" message concatenated from more than two simple messages has multiple term representation, depending on the order of concatenation.

We mechanized the most essential proofs: two crucial lemmas, availability and correctness of the $\mathcal{P}^{\text{B/T}}$ protocol and correctness of the (most involved) $\mathcal{P}^{\text{CR/TL}}$ protocol. The proofs follow the pen-and-paper proofs very closely. Each step of the pen-and-paper proof (i.e., each application of a feasibility rule) translates into an application of a number of Isabelle methods. The Isabelle source code for the model and proofs is roughly 2,500 lines long.

```
1: on NDstart(A; t1; B)
2:   Bcast(A; t1; ⟨A, t1, loc(A), auth_AB(t1, loc(A))⟩)
3: on Receive(B; t2; ⟨A, t1, l, auth_AB(t1, l)⟩)
4:   if t2 − t1 = d(loc(B), l)v^−1
5:     Neighbor(B; t2 + |A, t1, l, auth_AB(t1, l)|; A, B, t2)
```

Fig. 13. Pseudocode for protocol $\mathcal{P}^{\text{B/TL}}$.

```
01: on NDstart(A; t1; B)
02:   Fresh(A; t1 + |B|; n1)
03:   Bcast(A; t1; ⟨B, n1⟩)
04: on Receive(B; t; ⟨B, n1⟩)
05:   Bcast(B; t + Δ; ⟨auth_B(n1)⟩)
06: on Receive(A; t2; ⟨auth_B(n1)⟩)
07:   if occurred Fresh(A; t1 + |B|; n1)
08:   if occurred Bcast(A; t1; ⟨B, n1⟩)
09:   if v(t2 − t1 − Δ) ≤ 2R
10:     Neighbor(A; t2 + |auth_B(n1)|; A, B, t1)
11:     Neighbor(A; t2 + |auth_B(n1)|; B, A, t2)
```

Fig. 14. Pseudocode for protocol $\mathcal{P}^{\text{CR/T}}$.

5 Discussion

5.1 Modeling Assumptions

Mobility and NLOS delay. We assume nodes are static and NLOS delay is constant over time. This simplifies the model significantly; otherwise, propagation delay would vary during the transmission of a message. This is a reasonable assumption, because mobility and NLOS delay changes are very minor at the ND protocol execution time scale. For example, during 100 μs, nodes moving at

TABLE 1
Summary of Assumptions for Different ND Protocol Classes in Theorem 2

	Correctness	Availability
T-protocols		
B-based	$\Delta_{\text{relay}} > \frac{\mathbf{R}}{\mathbf{v}}$	$dist(A, B) \leq \mathbf{R}$
CR-based	$\Delta_{\text{relay}} > 2\frac{\mathbf{R}}{\mathbf{v}}$	$dist(A, B) \leq \mathbf{R}$
TL-protocols		
B-based	$\Delta_{\text{relay}} > 0, \mathbf{v} = \mathbf{v}_{\text{adv}}$	$nlos(A, B) = 0$
CR-based	$\Delta_{\text{relay}} > 0, \mathbf{v} = \mathbf{v}_{\text{adv}}$	$nlos(A, B) = 0$

100 kmph traverse 2.7 mm, which is below the accuracy of most RF ranging systems. However, in general, mobility can have security implications. To see why, consider the $\mathcal{P}^{\mathrm{CR/TL}}$ protocol. If nodes move during the protocol execution, it is important *when* they estimate their location. At the very least, A should estimate its location once when it sends the challenge, and again when it receives the response; whereas the responding node B should estimate its location when it sends the response. But even this might be insufficient under high mobility: If A measures its location at the beginning of the message, while B measures the ToF at the end of the message, there may be space for a stealthy relay attack. Introducing mobility and a dynamically changing NLOS delay in our model is a possible direction for future work.

Medium access control, jamming, directional antennas. For simplicity, we do not introduce any MAC restrictions into the model. Hence, a node is able to simultaneously receive any finite number of messages, even though in reality it is limited (to one message, or more for CDMA-like technologies). We could introduce additional rules that model radio interference, for example, set links *down* if two (or more, depending on the node transceiver capabilities) simultaneous transmissions take place. However, this would *not* affect any of our results. Notably, the availability properties require links to be *up*, but they are agnostic as to why links are *up* or *down*. Similarly, jamming would not affect our results either: we capture jamming with links being *down*, thus availability implies, among other things, no jamming. An adversary using directional antennas can also be modeled by links being *down*.

Inaccuracies. We assume correct nodes have accurate time and location information. However, in reality, inaccuracies are inevitable. Regarding time, clocks may be coarse grained and they can drift. Furthermore, there is always some error in estimating the message reception time (time-of-arrival) over a noisy channel. Regarding location, infrastructure (e.g., the global positioning system (GPS), or base stations) providing location information may be temporarily unavailable, and the location provided also includes some measurement error. Some of the inaccuracies can be decreased: For example, the error in message time-of-arrival can be decreased by averaging over many messages or introducing long physical-communication-layer preambles. But some inaccuracy in time and location is unavoidable.

Inaccuracies do not diminish the impossibility result; rather, they make it stronger. Indeed, we prove the impossibility result holds even in an idealized environment, in which nodes have access to information more accurate than in reality. In contrast, as secure ND protocols rely on distance estimates, their effectiveness can be negatively affected by such inaccuracies. For T-protocols, and even more so for TL-protocols, inaccuracies hinder availability: they can lead to ToF estimates seemingly above the threshold for T-protocols, and make the two distance estimates diverge for TL-protocols. The only way to cope with these is to introduce some tolerance margins for measurements. Nonetheless, this affects correctness: The higher the tolerance margin, the more space is left for fast relay attacks. This manifests the unsurprising tension between correctness and availability. Introducing inaccuracies explicitly into the framework is a possible direction for future work.

5.2 ND Protocol Comparison

T-protocols versus TL-protocols. On the one hand, TL-protocols provide stronger security than T-protocols in term of correctness (Table 1). First, they do not need the notion of ND range, $\mathbf{R}$, that T-protocols do. More important, they are secure as long as $\Delta_{\mathrm{relay}} > 0$. In contrast, T-protocols require that $\Delta_{\mathrm{relay}} \geq \mathbf{R}\mathbf{v}^{-1}$. On the other hand, TL-protocols suffer in terms of availability: 1) they require location-aware nodes with secure and precise location information, a far-from-trivial requirement, and 2) they do not work for links with substantial NLOS delay. In addition, TL-protocols require $\mathbf{v} = \mathbf{v}_{\mathrm{adv}}$ for correctness, which limits them to speed-of-light channels such as RF.

In light of these shortcomings, notably shortcoming 1, T-protocols can be a viable solution to provide communication ND, depending on the environment, the communication technology, and the sophistication of the adversary. First, T-protocols provide a good approximation of communication ND in environments without obstacles, although these are not very common in reality. Second, if the ND range $\mathbf{R}$ is low, than the adversary needs to be able to relay with a small Δ_{relay}. For example, if we consider relatively short-range IEEE 802.11 radios, with $\mathbf{R}$ in the order of 100 meters, $\Delta_{\mathrm{relay}} \approx 100\ \mathrm{mc}^{-1} \approx 333\ \mathrm{ns}$. This is significantly below the 15-20μs achievable by the relay constructed by Hancke in [12]. Simple store-and-forward relays are also thwarted easily. In contrast, for WiMAX, with a range up to 50 km, the lower-bound on Δ_{relay} is around $166\ \mu\mathrm{s}$ leaving much more space for attacks. In fact, as $\mathbf{R} \to \infty$, T-protocols become useless for securing ND.

In [10], Francillon et al. construct an analog relay with $\Delta_{\mathrm{relay}} \approx 20\ \mathrm{ns}$. A ND range $\mathbf{R}$ secure against such a relay is only a few meters. Furthermore, for many wireless technologies, 20 ns falls below the accuracy of the message time-of-arrival estimation. This implies that for practical purposes this relay could be assumed to achieve $\Delta_{\mathrm{relay}} \approx 0$, which would defeat not only T-protocols but nearly TL-protocols too. Furthermore, physical-communication-layer attacks [6] that construct a relay with a (seemingly) negative Δ_{relay}, would also defeat both classes of protocols.

Hence, it might appear that for a sophisticated enough adversary, communication ND is impossible not only for T-protocols, but also for TL-protocols. However, there is a significant difference between these two "impossibility results." For TL-protocols, the difficulty stems from the inaccuracy of time- and location measurements. These can be decreased by, for example, increasing the signal-to-noise ratio through making the message preambles (on which the time-of-arrival is estimated) longer. Furthermore, physical-communication-layer attacks can be mitigated with appropriate countermeasures [13], [25]. In contrast, the T-protocol impossibility is fundamental, and holds even in an idealized model with no inaccuracies and no physical layer attacks. For more on physical layer attacks, due to space limitations, please see the accompanying technical report, [26], Section 5.1.

B-protocols versus CR-protocols. B-protocols require (secure) clock synchronization, which limits their applicability, but they are more efficient. B-protocols involve fewer messages per protocol session; a group of n fully connected nodes only needs $O(n)$ protocol sessions versus $O(n^2)$ necessary for CR-protocols. Furthermore, B-protocols have less stringent availability requirements: links need be *up* for shorter periods than those needed by CR-protocols. Finally, in terms of correct (secure) operation, CR/T-protocols require Δ_{relay}, the minimum relaying delay, to be twice as large as that required by B/T-protocols (for the same $\mathbf{R}$).

Symmetric authenticators. Contrary to other protocols, the $\mathcal{P}^{\text{B/TL}}$ protocol uses a symmetric authenticator. For this reason, this protocol might seem at the first glance susceptible to a reflection attack: an adversarial node could receive a beacon message from node A and relay it back to A. The $\mathcal{P}^{\text{B/TL}}$ protocol is actually secure. Furthermore, we could modify all the other protocols by simply replacing asymmetric authenticators with symmetric ones, and they would still be secure under the same assumptions as their asymmetric counterparts. However, if we would remove the time and location information from the symmetric versions of the protocols, in an attempt to use them as regular authentication protocols, they would be all be vulnerable to the reflection attack. This demonstrates and interesting interplay between authentication and time/location features of ND protocols.

5.3 Beyond ND

In this paper, we focus on formal analysis of ND. However, our framework can be easily used to reason about other protocols, for example, for time synchronization or distance bounding (DB). The latter, allowing one device to estimate a secure upper bound on its distance to another device, have attracted considerable attention from the research community, especially their formal verification (see Section 6). However, there is a crucial reason why we refrain from reasoning about DB protocols. In ND protocols, traditionally, it is assumed that both participating nodes are honest. Under this assumption, no attacks that exploit manipulations on the bit-level have been discovered. Hence, our model is an adequate approximation for such protocols. In contrast, DB protocols assume that they can be executed with an adversarial node. If this is the case, an internal adversary is able to mount probabilistic bit-level attacks [6] that cannot be captured by our model and, in fact, by the majority of formal models developed for DB. There requires a new type of model that is probabilistic and models messages at the bit-level.

6 Related Work

A number of ND protocols can be found in the literature, but such schemes require the assistance of infrastructure or other nodes, and are hence not as generally applicable as the two-party ND we investigate in this paper. Among two-party ND, there are time-based approaches, notably the temporal packet leashes [15], and the TrueLink challenge-response protocol [9]; location-based approaches such as the geographical packet-leashes [15]; an approach based on using directional antenna [14], which provides relatively weak security guarantees in its basic two-part form; and a scheme based on device fingerprinting [29]. For a more comprehensive overview of ND schemes, we refer the reader to [20].

Impossibility results. In [5], the problem of secure clock synchronization under relay attacks turns out to be closely related to communication ND. Compared to our model, the model in [5] includes clock skews and an adversary model with the distinction of half-duplex, full-duplex, and double full-duplex transceivers, rather than the relaying delay. The authors obtain impossibility and possibility results for the considered transceiver types, which are complementary to the results obtained here. In contrast to our work, the authors abstract away the cryptographic aspect of the protocols—hence their framework cannot be directly use to prove the correctness of concrete protocols in the same way as our framework. This work is further extended in [30] to protocols for network-wide clock synchronization and topology discovery, but again cryptographic aspects are abstracting away.

Formal verification of protocols. A number of formal frameworks designed for verification of time-(and-location)-based protocols have been proposed. Contrary to our work, they tend to focus on *DB* rather than communication ND. DB protocols allow a node to establish a secure upper bound on the distance to another node. The two most prominent protocols of this type are the (original DB) Brands-Chaum [3] and the Hancke-Kuhn [11] protocols.

The first work where DB has been treated formally is [17] by Meadows et al. The authors extend existing formal approaches [4], [23] tailored to "classical" security protocols, and augments it with a notion of distance based on time stamps. It is not clear how communication neighborhood would be defined in this framework, nor how to model a protocol that uses location information.

In [31], Schaller et al. propose a framework based on the inductive active approach of Paulson [22]. The framework is formalized in the theorem prover Isabelle using Higher Order Logic. The authors use their framework to verify two DB protocols, as well as a delayed key disclosure protocol. They extend this approach in [2], proposing an elegant way of dealing with message spaces based on equational term theories. This, in particular, allows them to model the *exclusive or* (XOR) operation used in the Brands-Chaum protocols.

In [16], the authors extend the stand space model with timing information. The approach is automatized using the constrained solving techniques proposed in [18] for bounded-process analysis. The authors analyze four DB protocols, including a simplified version of the Brands-Chaum protocol, and a protocols proposed in [17]. They replace the XOR operation with symmetric encryption, and remove the commitment used in this protocol. The authors report that the constraint solver can efficiently find attacks in flawed versions of the protocols.

The authors of [33] extend the strand space formalism [32] with notions of message propagation time and device location to be able to reason about the security of simple DB and related protocols. No mechanization of the security proofs is provided.

Interestingly, in none of the above frameworks is it possible to prove the correctness of the (nonsimplified) Brands-Chaum protocol or the Hancke-Kuhn protocol [11]. One reason for this is the ability to model necessary "cryptographic" primitives: the combination of the XOR operation with the commitment in the Brands-Chaum protocol and the look-up operation in the Hancke-Kuhn protocol. More important, none of these frameworks models messages on the bit-level, and hence they miss the attack mentioned in Section 5.3. We believe these types of attacks should be possible to capture in any framework to prove the security of ND or DB protocols against internal adversaries. This would require a shift from nondeterministic, message-based models to a probabilistic bit-based models.

One formal approach takes this leap: The framework of Pavlovic and Meadows introduced in [24]. It includes probabilistic derivation based on the notion of guards, it models messages on the bit level, and it incorporates the Hancke-Kuhn look-up operation. In fact, [24] provides the first formal analysis of the Hancke-Kuhn protocol. However, the analysis and the paper has some limitations. The most important is that it is limited to two specific guessing attack scenarios (one distance fraud and one mafia fraud); nothing is mentioned about the terrorist fraud, to which the Hancke-Kuhn protocol is vulnerable. Such approaches, iterating over a list of high-level attack scenarios, offer limited security, because they ignore possibly undiscovered attack scenarios. An example of this limitation is the recently discovered distance hijacking attack scenario [7], which is different from the traditional distance fraud, mafia fraud, and terrorist fraud. In contrast, the strength of most formal approaches lies in considering a much broader scope of adversarial actions, and only restricting the adversary from violating causality, physical time and location constraints, and the security of cryptographic primitives. In [16], such an approach allowed the authors to detect a distance hijacking attack.

Compared to our framework, all the above-mentioned formal approaches focus on verification of concrete protocols. In some cases (e.g., [17], [31]), the authors move beyond the security of concrete protocols and show that all protocols of a particular form provide DB. However, there are no formal proofs of impossibility. Furthermore, our framework has another advantage (stemming from its impossibility-driven design). For example, in [31], the set of all "feasible traces" is the *least* set of traces in which correct nodes follow the protocol. This, strictly speaking means that the protocol is proven correct under the assumption that nodes do not execute any other protocol. In our approach, we consider the *greatest* set of traces in which correct nodes do not violate the protocol. Although this makes the protocol specification slightly more involved, it allows us to formalize a protocol in a way that can lead to stronger results, for example, we prove that a protocol $\mathcal{P}^{\mathrm{B/T}}$ can be safely composed with any other protocol that does not use messages of a particular form. Regarding composability, a somewhat similar alternative approach was proposed in [8].

7 Conclusions

We propose a formal framework for reasoning about the security of time-and-location-based protocols in wireless networks. The framework models aspects of wireless communications (neighbor relation, device locations, message propagation time). It is applicable to protocols such as DB, time synchronization, and most notably ND, which is the focus of this work.

We use the proposed framework to obtain two types of results. First, we show that the general class of time-based protocols can provide ND if and only if the adversary can only relay messages with a delay above a certain threshold (related to the desired communication range). We also show that time-and-location-based protocols offer superior security, as they can provide ND as soon as adversarial relaying delay is strictly positive. However, these protocols only work for speed-of-light channels and if there is no NLOS propagation delay. Second, we use the framework to prove the security of concrete ND protocols, including two novel time-and-location-based protocols. We mechanize the model and proofs in the theorem prover Isabelle, for a high level of guarantee on the security of these protocols.

We argue that the proposed framework is adequate for reasoning about ND (and related protocols) under the assumption that both participating nodes are honest. However, if one of the nodes is allowed to be adversarial—a common assumption for DB protocols—then this opens a whole new range of probabilistic bit-level attacks, as well as attacks on the physical-communication-layer. This mandates, in our opinion, a shift from (non)deterministic message-oriented models to probabilistic models that explicitly consider bits or even symbols at the physical-communication-layer. We believe this to be an interesting direction for future work.

References

[1] Nokia Instant Community Project, http://research.nokia.com/news/9391, 2013.

[2] D. Basin, S. Čapkun, P. Schaller, and B. Schmidt, "Let's Get Physical: Models and Methods for Real-World Security Protocols," *Proc. 22nd Int'l Conf. Theorem Proving in Higher Order Logics (TPHOLs)*, 2009.

[3] S. Brands and D. Chaum, "Distance-Bounding Protocols," *Proc. Advances in Cryptology, Workshop Theory and Application of Cryptographic Techniques (EUROCRYPT)*, 1994.

[4] I. Cervesato, C. Meadows, and D. Pavlovic, "An Encapsulated Authentication Logic for Reasoning about Key Distribution Protocols," *Proc. IEEE 18th Workshop Computer Security Foundations*, 2005.

[5] J. Chiang, J. Haas, Y.-C. Hu, P. Kumar, and J. Choi, "Fundamental Limits on Secure Clock Synchronization and Man-in-the-Middle Detection in Fixed Wireless Networks," *Proc. IEEE INFOCOM*, 2009.

[6] J. Clulow, G.P. Hancke, M.G. Kuhn, and T. Moore, "So Near and Yet So Far: Distance-Bounding Attacks in Wireless Networks," *Proc. Third European Workshop Security and Privacy in Ad Hoc and Sensor Networks (ESAS)*, 2006.

[7] C. Cremers, K.B. Rasmussen, and S. Capkun, "Distance Hijacking Attacks on Distance Bounding Protocols," *Proc. 2012 IEEE Symposium on Security and Privacy*, 2012.

[8] A. Datta, A. Derek, J. Mitchell, and D. Pavlovic, "A Derivation System and Compositional Logic for Security Protocols," *J. Computer Security*, vol. 13, pp. 423-482, 2005.

[9] J. Eriksson, S.V. Krishnamurthy, and M. Faloutsos, "TrueLink: A Practical Countermeasure to the Wormhole Attack in Wireless Networks," *Proc. IEEE 14th Int'l Conf. Network Protocols (ICNP)*, 2006.

[10] A. Francillon, B. Danev, and S. Čapkun, "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars," *Proc. 18th Ann. Network and Distributed System Security Symp. (NDSS)*, 2011.
[11] G. Hancke and M. Kuhn, "An RFID Distance Bounding Protocol," *Proc. First Int'l Conf. Security and Privacy for Emerging Areas in Comm. Networks (SecureComm)*, 2005.
[12] G.P. Hancke, "Practical Attacks on Proximity Identification Systems," *Proc. IEEE Symp. Security and Privacy*, 2006.
[13] G.P. Hancke and M.G. Kuhn, "Attacks on Time-of-Flight Distance Bounding Channels," *Proc. First ACM Conf. Wireless Network Security (WiSec)*, 2008.
[14] L. Hu and D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks," *Proc. 11th Ann. Symp. Network and Distributed Systems Security (NDSS)*, 2004.
[15] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks," *Proc. IEEE INFOCOM*, 2003.
[16] S. Malladi, B. Bezawada, and K. Kothapalli, "Automatic Analysis of Distance Bounding Protocols," *Proc. Workshop Foundations of Computer Security*, 2009.
[17] C. Meadows, R. Poovendran, D. Pavlovic, L.-W. Chang, and P. Syverson, "Distance Bounding Protocols: Authentication Logic Analysis and Collusion Attacks," *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*. Springer, 2007.
[18] J. Millen and V. Shmatikov, "Constraint Solving for Bounded Process Cryptographic Protocol Analysis," *Proc. Eighth ACM Conf. Computer and Comm. Security (CCS)*, 2001.
[19] T. Nipkow, L.C. Paulson, and M. Wenzel, *A Proof Assistant for Higher Order Logic*. Springer, 2008.
[20] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Čapkun, and J.-P. Hubaux, "Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networking," *IEEE Comm. Magazine*, vol. 46, no. 2, pp. 132-139, Feb. 2008.
[21] P. Papadimitratos and Z.J. Haas, "Secure Message Transmission in Mobile Ad Hoc Networks," *Ad Hoc Networks J.*, vol. 1, no. 1, pp. 193-209, 2003.
[22] L.C. Paulson, "The Inductive Approach to Verifying Cryptographic Protocols," *J. Computer Security*, vol. 6, no. 1/2, pp. 85-128, 1998.
[23] D. Pavlovic and C. Meadows, "Deriving Secrecy Properties in Key Establishment Protocols," *Proc. 11th European Symp. Research in Computer Security (ESORICS)*, 2006.
[24] D. Pavlovic and C. Meadows, "Bayesian Authentication: Quantifying Security of the Hancke-Kuhn Protocol," *Proc. 26th Conf. Math. Foundations of Programming Semantics (MFPS)*, 2010.
[25] M. Poturalski, M. Flury, P. Papadimitratos, J.-P. Hubaux, and J.-Y. Le Boudec, "Distance Bounding with IEEE 802.15.4a: Attacks and Countermeasures," *IEEE Trans. Wireless Comm.*, vol. 10, no. 4, pp. 1334-1344, Apr. 2011.
[26] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Formal Analysis of Secure Neighbor Discovery in Wireless Networks," http://www.ee.kth.se/~papadim/publications/fulltext/TECH_REPORT_SND_V2.pdf, technical report, 2012.
[27] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility," *Proc. Third ACM Symp. Information, Computer and Comm. Security (ASIACCS)*, 2008.
[28] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Towards Provable Secure Neighbor Discovery in Wireless Networks," *Proc. Sixth ACM Workshop Formal Methods in Security Eng.*, 2008.
[29] K.B. Rasmussen and S. Čapkun, "Implications of Radio Fingerprinting on the Security of Sensor Networks," *Proc. IEEE Third Conf. Security and Privacy in Comm. Networks (SecureComm)*, 2007.
[30] R. Robles, J. Haas, J. Chiang, Y.-C. Hu, and P. Kumar, "Secure Topology Discovery through Network-Wide Clock Synchronization," *Proc. Third Int'l Conf. Signal Processing and Comm. (SPCOM)*, 2010.
[31] P. Schaller, B. Schmidt, D. Basin, and S. Čapkun, "Modeling and Verifying Physical Properties of Security Protocols for Wireless Networks," *Proc. IEEE 22nd Computer Security Foundations Symp. (CSF)*, 2009.
[32] F.J. Thayer, J.C. Herzog, and J.D. Guttman, "Strand Spaces: Why Is a Security Protocol Correct?" *Proc. IEEE Symp. Security and Privacy*, 1998.
[33] F.J. Thayer, V. Swarup, and J.D. Guttman, "Metric Strand Spaces for Locale Authentication Protocols," *Proc. Fourth IFIP Int'l Conf. Trust Management*, 2010.

Marcin Poturalski received the PhD degree from the School of Computer and Communication Sciences, the Swiss Federal Institute of Technology in Lausanne, in 2011. His research interests include the security of wireless communication, notably neighbor discovery, ranging, and localization. He is a student member of the IEEE.

Panos Papadimitratos received the PhD degree from Cornell University, Ithaca, New York, in 2005. He is currently an associate professor in the School of Electrical Engineering at the Royal Institute of Technology, Stockholm, Sweden. His research is concerned with security and networked systems. He is a member of the IEEE.

Jean-Pierre Hubaux joined the faculty of the Swiss Federal Institute of Technology in Lausanne in 1990. He became a full professor in 1996. He held visiting positions at the IBM T.J. Watson Research Center and at the University of California, Berkeley. He is one of the seven commissioners of the Swiss Federal Communications Commission. His current research is focused on privacy preservation mechanisms in pervasive communications. He is a fellow of both the ACM and the IEEE.