Modeling and Verifying Physical Properties of Security Protocols for Wireless Networks

Patrick Schaller, Benedikt Schmidt, David Basin, Srdjan Capkun
ETH Zurich, Switzerland
Email: {patrick.schaller,benedikt.schmidt,david.basin,srdjan.capkun}@inf.ethz.ch

Abstract

We present a formal model for modeling and reasoning about security protocols. Our model extends standard, inductive, trace-based, symbolic approaches with a formalization of physical properties of the environment, namely communication, location, and time. In particular, communication is subject to physical constraints, for example, message transmission takes time determined by the communication medium used and the distance traveled. All agents, including intruders, are subject to these constraints and this results in a distributed intruder with restricted, but more realistic, communication capabilities than those of the standard Dolev-Yao intruder. We have formalized our model in Isabelle/HOL and used it to verify protocols for authenticated ranging, distance bounding, and broadcast authentication based on delayed key disclosure.

1. Introduction

The shrinking size of microprocessors combined with the ubiquity of wireless network connections has led to new application areas for networked systems with novel security requirements for the protocols employed. Whereas traditional security protocols are mainly concerned with message secrecy or variants of authentication, new application areas often call for new protocols that securely establish properties of the network environment. Examples include:

Physical Proximity: One node must prove to another node that a given value is a reliable upper bound on the physical distance between them. Such protocols may use authentication patterns along with assumptions about the underlying communication medium, e.g., [9], [11], [21], [26].

Secure Localization: A node must determine its true location in an adversarial setting or make verifiable statements about its location by executing protocols with other nodes, e.g., [10], [24], [25], [36], [39]. Secure localization and physical proximity verification protocols, and attacks on them, have been implemented on RFID, smart cards, and Ultra-Wide Band platforms [17], [35].

Secure Time Synchronization: A node must securely synchronize its clock to the clock of another (trusted) node in an adversarial setting, e.g., [19], [40]. These protocols also serve as a basis for efficient secure networking protocols, e.g., for efficient broadcast authentication [31].

Secure Neighbor Discovery or Verification: A node must determine or verify its direct communication partners within a communication network [29]. Correct information about the network topology is essential for all routing protocols.

What these examples have in common is that they all concern physical properties of the communication medium or the environment in which the nodes live. Furthermore, all of these protocols fall outside the scope of standard symbolic protocol models based on the Dolev-Yao intruder.[1]

In this paper, we present a formal model for reasoning about the security guarantees of protocols like those listed above. Our model builds on standard symbolic approaches and accounts for physical properties like time, the location of network nodes, and properties of the communication medium. Honest agents and the intruder are modeled as network nodes. The intruder, in particular, is not modeled as a single entity but rather a distributed one and therefore corresponds to a set of nodes. The ability of the nodes to communicate and the speed of communication are determined by nodes' locations and by the propagation delays of the communication technologies they use. As a consequence, nodes (both honest and those controlled by the intruder) require time to share their knowledge and information cannot travel between nodes at speeds faster than the speed of light. The intruder and honest agents are therefore subject to physical restrictions. This results in a distributed intruder with communication abilities that are restricted, but more realistic than the classical Dolev-Yao intruder.

Our model combines a message and a communication model. Whereas the message model allows us to capture cryptographic aspects of protocol messages (under the assumption of perfect cryptography), our communication model allows us to model relevant properties of the communication technology. Similar to Paulson's *Inductive Approach* [30], we have used Isabelle/HOL [28] to formalize our model and to prove security properties of the protocols

[1] This is understandable: the Dolev-Yao model was developed for classical security protocols, whose correctness is independent of the details of the physical environment. Abstracting these details away by identifying the network with the intruder results in a simpler model and can also be motivated as modeling a strong intruder who controls the entire network.

presented in this paper. We model communication as send and receive events, where the communication technology and the network topology determine the time and location of the receive event resulting from a given send event.

As applications, we have formalized and verified three protocols. Their diverse features and properties reflect the broad scope of our model in applications where environmental factors and their physical constraints are used alongside cryptography to achieve security objectives.

The rest of the paper is organized as follows. In Section 2, we present an example protocol and background on protocol formalization and Isabelle/HOL. In Section 3, we present our model. In Section 4, we describe the protocols that we formalize and the proofs of their security properties. In Section 5, we discuss our formalization [6], before surveying related work and drawing conclusions in Sections 6 and 7.

2. Background

2.1. An Example: Authenticated Ranging

As an example of a physical-proximity protocol, we present a version of an *authenticated ranging* protocol, shown in Figure 1 (see [9], [10] for details on authenticated ranging).

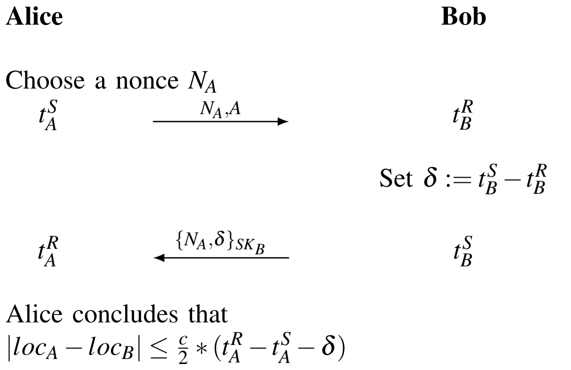

Figure 1: Authenticated Ranging Protocol

The protocol's objective is for the verifier (Alice) to determine a reliable upper bound on the distance to the trusted prover (Bob) in an adversarial environment. To achieve this, Alice uses her knowledge about the communication technology that she and Bob use to exchange information. She uses the protocol to measure the round-trip time-of-flight of a signal (traveling at the speed of light c) between her and Bob. In particular, she creates a fresh, unguessable nonce and sends it to Bob at time t_A^S. After receiving the nonce, Bob concatenates it with the processing time δ (the time between receiving the nonce and sending his response) and signs the message with his private key SK_B to prove that the message originates from him. Upon receiving the reply, Alice notes the time of reception t_A^R and calculates the time-of-flight, $t_A^R - t_A^S - \delta$. Since the computation time is used in the calculation of the distance, the prover Bob (the owner of the signing key used) must be trusted. As an application for such a protocol, imagine a door-locking system that requires that a legitimate key, such as an RFID card, must be close to a door for the door's lock to open.

This simple example shows how nodes can combine time, relative location, and properties of the communication medium, together with cryptographic functionality to securely deduce properties of their physical environment. Any formal model intended to reason about such protocols must therefore take such physical properties into account.

2.2. The Inductive Approach

We formalize our model within higher-order logic in the Isabelle/HOL system, extending the inductive approach to security protocol verification introduced by Paulson in [30]. This approach is based on a trace-based interleaving semantics, which gives a semantics to distributed systems as the set of traces describing all possible interleaved agent executions. In particular, protocols are modeled by rules describing the protocol steps executed by honest agents and possible intruder actions. The set of rules constitutes an inductive definition that defines the protocol's semantics as an infinite set of communication traces, each trace being a finite list of communication events. Security properties are also specified as sets of traces, usually defined by predicates on traces. Protocol security is then reduced to language containment: a protocol is secure relative to a property (predicate) if the property holds for all protocol traces. This is proved by induction on traces using an induction principle derived from the protocol rules.

2.3. Isabelle/HOL

Isabelle [28] is a generic theorem prover with a specialization for higher-order logic (HOL). We will avoid Isabelle-specific details as much as possible or explain them in context, as needed.

Here we limit ourselves to few comments on typing. A function f from type α to β is denoted $f : \alpha \to \beta$ and $c\,x \equiv t$ defines the function c with parameter x as the term t. We write $\alpha \times \beta$ for the product type of α and β and we use the predefined list type α *list* with the append ($xs.x$) operation. Algebraic data types can be defined using the **datatype** declaration.

Central to our work is the ability to define (parameterized) inductively defined sets. These sets are defined by sets of rules and denote the least set closed under the rules. Given an inductive definition, Isabelle generates a rule for proof by induction. Examples of this and datatype definitions are provided in Section 3.

3. Formal Model

In this section, we present our model, which incorporates node location, time, and a notion of communication distance. Before presenting the technical details, we introduce the concepts modeled.

3.1. Concepts Modeled

Agents. We consider a set of communicating agents, consisting of honest and dishonest agents. Honest agents follow the protocol rules, whereas dishonest agents (also called intruders) can deviate arbitrarily from the protocol. Each agent has a fixed location, and a set of transmitters and receivers. Agents can have initial knowledge (such as their own private keys and the public keys of other agents), which they use to construct new messages or to analyze intercepted messages.

Network. We model an unreliable network connecting agents' transmitters and receivers as a matrix. The matrix describes the connectivity between transmitters and receivers, whereby an agent Alice can send messages directly to an agent Bob if and only if Alice is connected to Bob in the matrix. The matrix entries express the lower bounds on the signal propagation time from a transmitter to a receiver. They therefore formalize not only whether direct communication is possible, but also the effect of different communication technologies with different signal propagation velocities, e.g., radio and ultrasound transmission. Modeling an unreliable network allows us to capture message deletion (jamming) and transmission failures.

Our model distinguishes between the topology associated with the agents' locations and the topology associated with the network. Whereas physical distance corresponds to Euclidean distance, the network topology describes signal paths not necessarily corresponding to the line-of-sight paths between senders and receivers (e.g., rolled up cables, signal reflections). However, to accurately model reality, the communication model must be consistent with basic physical laws. In particular, the smallest transmission time possible between transmitters and receivers corresponds to line-of-sight transmission.

Time. Protocols, such as the example from Section 2.1, measure and make statements about time. As a result, our model must correctly describe temporal dependencies between related events, such as a send event preceding a receive event and agents must be able to access clocks to associate events with timestamps. We realize this by tagging every event with the corresponding timestamp. We model temporal dependencies and clock access by agents using inductive rules that account for arbitrary offsets of local clocks.

Intruder Model. In most formal approaches to security protocol analysis, the intruder is modeled as a single entity, following the Dolev-Yao intruder [16]. This intruder can defy the laws of physics by simultaneously observing all network traffic, an abstraction that is reasonable for reasoning about protocols involving properties not dependent on time or distance. Moreover, cryptography is modeled as a black box (the perfect cryptography assumption) where the intruder can construct messages in different predefined ways, but he cannot break cryptography. For example he cannot decrypt a message without an appropriate key. In our model, we also employ the perfect cryptography assumption.

We model message exchanges between the agents taking into account their communication distance, as specified by the network communication matrix. The constraints on communication apply both to honest agents and intruders. An individual intruder can therefore only intercept messages at his location. Moreover, colluding intruders cannot instantaneously exchange knowledge. They must exchange messages using the network topology, as defined by the communication matrix. This models reality, where the attackers' ability to observe and communicate messages is determined by their locations, mutual distances, and by their transmitters and receivers.

3.2. Agents and the Environment

We now present our model and sketch its formalization in Isabelle/HOL (see [6] for details).

Agents and Transmitters. Agents are either honest or dishonest (intruders). We model each kind using the set of natural numbers *nat* and hence there are infinitely many agents of each kind.

$$\textbf{datatype}\ \mathit{agent} = \mathsf{Honest}\ \mathit{nat} \mid \mathsf{Intruder}\ \mathit{nat}$$

We refer to agents using capital letters like A and B. We also write H_A and H_B for honest agents and I_A and I_B for intruders, when we require this distinction. Each agent has a set of transmitters and receivers.

$$\textbf{datatype}\ \mathit{transmitter} = \mathsf{Tx}\ \mathit{agent}\ \mathit{nat}$$

The constructor Tx returns a transmitter, given an agent A and an index i, denoted Tx_A^i. The number of usable transmitters can be restricted by specifying that some transmitters cannot communicate with any receivers. Receivers are formalized analogously.

$$\textbf{datatype}\ \mathit{receiver} = \mathsf{Rx}\ \mathit{agent}\ \mathit{nat}$$

Physical and Communication Distance. The function loc assigns each agent A a location $loc_A \in \mathbb{R}^3$. Using the standard Euclidean metric on $\mathbb{R}^3$, we define the physical distance between two agents A and B as $|loc_A - loc_B|$.

Taking the straight-line distance between the locations of the agents A and B in $\mathbb{R}^3$ as the shortest path (taken for example by electromagnetic waves when there are no obstacles), we define the line-of-sight communication distance as:

$$cdist_{LoS}(A,B) = \frac{|\,loc_A - loc_B\,|}{c},$$

where c is the speed of light.

The value computed by $cdist_{LoS}$ only depends on A and B's and is independent of the network topology. We model the network topology using the function $cdist_{Net}$: $transmitter \times receiver \rightarrow \mathbb{R}_{\geq 0} \cup \{\perp\}$, whose value depends on the communication medium used by the given transceivers, obstacles between transmitters and receivers, and other environmental factors. $cdist_{Net}(Tx_A^i, Rx_B^j) = \perp$ denotes that Rx_B^j cannot receive transmissions from Tx_A^i. In contrast, $cdist_{Net}(Tx_A^i, Rx_B^j) = t$, where $t \neq \perp$, describes that Rx_B^j may receive signals (messages) emitted by Tx_A^i after a delay of t time units. Since we assume that information cannot propagate faster than with the speed of light, we always require that

$$cdist_{LoS}(A,B) \leq cdist_{Net}(Tx_A^i, Rx_B^j).$$

In Isabelle/HOL, we model loc as an uninterpreted function constant and define $cdist_{LoS}$ in terms of loc. The function $cdist_{Net}$ is also uninterpreted, but is required to have the previously mentioned property: faster-than-light communication is impossible. Additional assumptions about the agents' locations and the network topology needed for analyzing protocols can be added as local assumptions in security proofs. Hence, our results apply to all possible locations of agents and to all network topologies that fulfill the assumptions.

Relation between the two Notions of Distance. The following example relates communication and physical distance. The left side of Figure 2 illustrates the nodes and their environment. Here, edges denote line-of-sight connections (shortest paths in Euclidean space) and are labeled with the corresponding values of the $cdist_{LoS}$ function. Note that $cdist_{LoS}$ is defined in terms of the physical location of nodes and neither depends on communication obstacles nor physical properties of the communication medium.

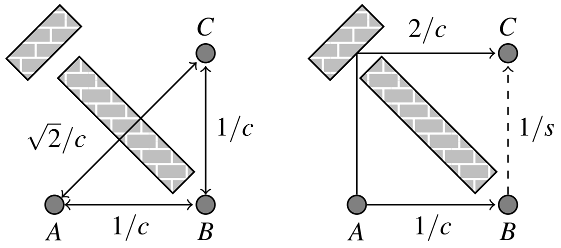

Figure 2: Physical (left) and Network Topology (right).

The right side of Figure 2 illustrates the communication distance associated with the network topology. The dashed line here represents an ultrasonic link, where signals travel at the speed of sound s. The long wall in the middle prevents line-of-sight communication from A to C. However, reflection off the short wall enables C to receive the signal. So the two notions of distance only coincide for the link from A to B, which uses line-of-sight communication at the speed of light c.

3.3. Messages and Events

Messages. A message is either atomic or composed. Atomic messages are agent names, times, nonces, numbers, and keys. Composed messages are hashes, pairs, and encrypted messages.

$$\begin{aligned}\textbf{datatype}\ msg = {} & \mathsf{Agent}\ agent & \mid\ & \mathsf{Time}\ real\\ \mid\ & \mathsf{Number}\ nat & \mid\ & \mathsf{Nonce}\ agent\ nat\\ \mid\ & \mathsf{Key}\ key & \mid\ & \mathsf{Hash}\ msg\\ \mid\ & \mathsf{MPair}\ msg\ msg & \mid\ & \mathsf{Crypt}\ key\ msg\end{aligned}$$

Nonces are tagged with the name of the agent who created them and a unique identifier. This ensures that nonces created by different agents never collide. Indeed, even colluding intruders must communicate to share a nonce. The freshness of nonces (introduced at a given event in a trace) is guaranteed at creation time by the predicate *used* introduced below. Similar to nonces, keys are assigned a unique value, whereby the set of keys is partitioned into those used for asymmetric encryption and symmetric encryption. An inverse operator $\cdot^{-1}$ is defined for both key types (it is the identity function on symmetric keys). The constructor *Crypt* denotes signing, asymmetric, or symmetric encryption, depending on the key used. We write $\{m\}_k$ for $Crypt\ k\,m$ and (m,n) for $MPair\ m\,n$.

Given a set of messages, an agent can derive new messages by decomposing and composing given messages. We formalize this message derivation capability with the inductively defined operator $DM : agent \ \rightarrow msg\ set \ \rightarrow msg\ set$. The rules comprising DM are listed in Figure 3 and specify message decryption, projection on pairs, pairing, encryption, signing, hashing, and the generation of numbers, time values, agent names, and nonces. For example, the *Dec*-rule states that if an agent A can derive the ciphertext $\{m\}_k$ and the decryption key $(Key\ k)^{-1}$, then he can also derive the cleartext m. When $Key\ k$ is used as a signing key, A uses the verification key $(Key\ k)^{-1}$ to verify the signature.

Events and Traces. An event corresponds to an agent taking one of the three actions: sending or receiving a message or making a claim.

$$\begin{aligned}\textbf{datatype}\ event = {} & \mathsf{Send}\ transmitter\ msg\\ \mid\ & \mathsf{Recv}\ receiver\ msg\\ \mid\ & \mathsf{Claim}\ agent\ msg\end{aligned}$$

A *trace* is a list of timed events, where timed events $(t,e) \in real \ \times event$ pair a timestamp with an event. A

$$\frac{m \in M}{m \in DM_A(M)}\ \textsc{inj} \qquad \frac{m \in DM_A(M)}{Hash\ m \in DM_A(M)}\ \textsc{hash}$$

$$\frac{(m,n) \in DM_A(M)}{m \in DM_A(M)}\ \textsc{fst} \qquad \frac{(m,n) \in DM_A(M)}{n \in DM_A(M)}\ \textsc{snd}$$

$$\frac{m \in DM_A(M) \qquad n \in DM_A(M)}{(m,n) \in DM_A(M)}\ \textsc{pair}$$

$$\frac{m \in DM_A(M) \qquad Key\ k \in DM_A(M)}{\{m\}_k \in DM_A(M)}\ \textsc{enc}$$

$$\frac{\{m\}_k \in DM_A(M) \qquad (Key\ k)^{-1} \in DM_A(M)}{m \in DM_A(M)}\ \textsc{dec}$$

$$\frac{}{Time\ t \in DM_A(M)}\ \textsc{time} \qquad \frac{}{Agent\ a \in DM_A(M)}\ \textsc{agent}$$

$$\frac{}{Number\ n \in DM_A(M)}\ \textsc{number}$$

$$\frac{}{Nonce\ A\ n \in DM_A(M)}\ \textsc{nonce}$$

Figure 3: Rules for $DM_A(M)$

timed event $(t^S, Send\ Tx_A^i\ m)$ denotes that agent A has sent a message m using his transmitter Tx_A^i at time t^S. Such a *Send*-event may result in multiple *Recv*-events of the form $(t^R, Recv\ Rx_B^j\ m)$, where the timestamps t^R and the receivers Rx_B^j are consistent with the network topology.

A *Claim*-event models a belief or conclusion made by a protocol participant, formalized as a message. For example, after successfully completing a run of the authenticated ranging protocol (see Section 2.1) with Bob, Alice concludes at time t^C ($t_A^R \leq t^C$) that d_{AB} is an upper bound on her distance to Bob. We model this by adding the event $(t^C, Claim\ A\ (B, d_{AB}))$ to the trace. The protocol is therefore secure if the claim about the upper bound on the mutual distance holds for all traces containing such a claim event, where the protocol is used in an environment consistent with the model (as defined by *loc* and $cdist_{Net}$).

Knowledge and Used Messages. Each agent A possesses some initial knowledge, denoted $initKnows_A$ which depends on the concrete protocol to be executed. In a system run with trace tr, knowledge is defined as the union of the initial knowledge and all received messages.

$$\begin{aligned} knows_A(tr) \equiv \{m \mid \exists k\, t.(t,\ Recv\ Tx_A^k\ m) \in tr\} \\ \cup\ initKnows_A \end{aligned}$$

Each agent can derive all messages in the set $DM_A(knows_A(tr))$ by applying the derivation operator to the set of known messages.

All subterms n of m, excluding those that only appear as keys in *Crypt* or as messages in *Hash*, are parts of m, written as $n \sqsubseteq m$. We use this to define the set of messages used in a trace tr.

$$used(tr) \equiv \{n \mid \exists A\ k\ t\ m.(t, Send Tx_A^k\ m) \in tr\ \wedge\ n \sqsubseteq m\}$$

We say a message m originates at an event a_i in a trace $tr = [a_1, \ldots, a_{i-1}, a_i, \ldots, a_n]$, if $m \notin used([a_1, \ldots, a_{i-1}])$ and $m \in used([a_1, \ldots, a_i])$. In other words, a_i is the first event that uses m.

3.4. Network, Intruder, and Protocols

We now describe the inductive rules defining the set of traces $Tr(proto)$ for a system parameterized by a protocol *proto*. The base case, modeled by the NIL rule, states that the empty trace is a valid trace for all protocols. The other rules describe how a valid trace can be extended. They model the network behavior, the possible actions of the intruders, and the actions taken by honest agents following the protocol steps.

Network Rule. The NET-rule models message transmission from transmitters to receivers, constrained by the network topology described by $cdist_{Net}$. A *Send*-event from a transmitter may induce a *Recv*-event at a receiver only if the receiver can receive messages from the transmitter as specified by $cdist_{Net}$. The time delay between the related events is bounded below by the communication distance between the transmitter and receiver.

$$\frac{\begin{array}{c} tr \in Tr(proto) \quad t^R \geq maxtime(tr) \\ (t^S, Send\ Tx_A^i\ m) \in tr \quad cdist_{Net}(Tx_A^i, Rx_B^j) = t_{AB} \\ t_{AB} \neq \bot \quad t^R \geq t^S + t_{AB} \end{array}}{tr.(t^R, Recv\ Rx_B^j\ m) \in Tr(proto)}\ \textsc{Net}$$

If there is a *Send*-event in the trace tr and the premises of the NET-rule are fulfilled, a corresponding *Recv*-event is appended to the trace. The restriction on the connectivity and the transmission delay are ensured by $t_{AB} \neq \bot$ and $t^R \geq t^S + t_{AB}$. Here, t_{AB} is the communication distance between the receiver and the transmitter, t^S is the sending time, and t^R is the receiving time.

Note that a given *Send*-event can result in an unlimited number of *Recv*-events at the same receiver at different times. This is because $cdist_{Net}$ models the minimal communication distance and messages may also arrive later, for example due to reflection of the signal carrying the message. In addition, a *Send*-event can result in multiple *Recv*-events at different receivers, modeling for example broadcast communication. Finally, note that transmission failures and jamming by the intruder resulting in message loss are captured by not applying the NET-rule for a given *Send*-event and receiver, even if all premises are fulfilled.

We model message transmission with atomic *Send*-events and *Recv*-events. The timestamps associated with these events denote the starting times of message transmission and reception. Thus, our network rule captures the latency of the link, but not the message transmission time, which depends on the message's size and the transmission speed of transmitter and receiver. Some implementation specific attacks, for example as described in [13] and [36], are therefore not captured in our model. As future work, we plan to enrich our model to capture such attacks as well.

The premise $t \geq \mathit{maxtime}(tr)$, included in every rule (except the NIL-rule), ensures monotonically increasing timestamps in all traces. Here t denotes the timestamp associated with the new event and $\mathit{maxtime}(tr)$ denotes the latest timestamp of trace tr. This guarantees that the partial order on timed events induced by the timestamps (note that events can happen simultaneously) is consistent with the order of events in the list representing the trace.

Intruder Rule. The FAKE-rule describes the intruders' behavior. Namely, an intruder can always send any message derivable from his knowledge.

$$\frac{\begin{array}{c} tr \in \mathit{Tr}(\mathit{proto}) \quad t \geq \mathit{maxtime}(tr) \\ m \in \mathit{DM}_{I_A}(\mathit{knows}_{I_A}(tr)) \end{array}}{tr.(t, \mathit{Send}(\mathit{Tx}^k_{I_A}, m)) \in \mathit{Tr}(\mathit{proto})} \ \textsc{Fake}$$

Since knowledge is distributed, we use explicit *Send*-events and *Recv*-events to model the exchange of information between colluding intruders. With an appropriate $\mathit{cdist}_{\mathit{Net}}$ function, it is possible to model an environment where the intruders are connected by high-speed links, allowing them to carry out wormhole attacks. Restrictions on the degree of cooperation between intruders can be modeled as predicates on traces. Internal and external attackers are both captured since they differ only in their initial knowledge (or associated transceivers), which can be defined accordingly.

Protocols. In contrast to intruders who can send arbitrary derivable messages, honest agents follow the protocol. A protocol is defined by a set of step functions. Each step function takes the local view and time of an agent as input and returns all possible actions compliant with the protocol specification.

There are two types of possible actions modeling either a *Send*-event with a given transmitter id or a *Claim*-event.

$$\textbf{datatype}\ \mathit{action} = \ \mathsf{SendA}\ \mathit{nat} \mid \mathsf{ClaimA}$$

An *action* associated with an agent and a message can be translated into the corresponding trace event using the *translateEv* function.

$$\begin{array}{r} \mathit{translateEv}(A, \mathsf{SendA}\ \ k, m) \equiv \mathit{Send}\ \mathit{Tx}^k_A\ \ m \\ \mathit{translateEv}(A, \mathsf{ClaimA}\ , m) \equiv \mathit{Claim}\ A\ m \end{array}$$

A protocol step is therefore of type $\mathit{agent} \times \mathit{trace}\ \times \mathit{real} \rightarrow (\mathit{action}\ \times \mathit{msg})\ \mathit{set}$.

Since the actions of an agent A only depend on his own previous actions and observations, we define A's view of a trace tr as the projection of tr on those events involving A. For this purpose, we introduce the function *occursAt*, which maps events to associated agents, e.g. $\mathit{occursAt}(\mathit{Send}\ \mathit{Tx}^i_A\ \ m) \equiv A$.

$$\begin{array}{l} \mathit{view}(A, tr) \equiv [(\mathit{ctime}(A, t), ev) \mid \\ \qquad\qquad\qquad (t, ev) \in tr \wedge \mathit{occursAt}(ev) = A] \end{array}$$

Since timestamps of trace events refer to the global clock, the *view* function accounts for the offset of A's clock by translating times using the *ctime* function. Given an agent and a global time, the uninterpreted function $\mathit{ctime} : \mathit{agent} \times \mathit{real}\ \rightarrow \mathit{real}$ returns the corresponding time for the agent's clock.

Using the previous definitions, we define the PROTO rule. For a given protocol, specified as a set of the step functions, the PROTO rule describes all possible actions of honest agents, given their local views of a valid trace at a given point in time. If all premises are met, the PROTO-rule appends the translated event to the trace. Note that agents' behavior, modeled by the *step* function, is based only on the local clocks of the agents, i.e., agents cannot access global time.

$$\frac{\begin{array}{c} tr \in \mathit{Tr}(\mathit{proto}) \quad t \geq \mathit{maxtime}(tr) \\ \mathit{step} \in \mathit{proto} \\ (\mathit{act}, m) \in \mathit{step}(\mathit{view}(H_A, tr), H_A, \mathit{ctime}(H_A, t)) \\ m \in \mathit{DM}_{H_A}(\mathit{knows}_{H_A}(tr)) \end{array}}{tr.(t, \mathit{translateEv}(H_A, \mathit{act}, m)) \in \mathit{Tr}(\mathit{proto})} \ \textsc{Proto}$$

The restriction that all messages must be in $\mathit{DM}_{H_A}(\mathit{knows}_{H_A}(tr))$ ensures that agents only send messages derivable from their knowledge. This is the case for all executable protocols.

For a protocol *proto* given as a set of step functions, the set of all possible traces $\mathit{Tr}(\mathit{proto})$ is inductively defined by the NIL, NET, FAKE, and PROTO rules.

3.5. Protocol independent results

Since the set of traces $\mathit{Tr}(\mathit{proto})$ is parameterized by the protocol description *proto*, our model allows us to establish protocol-independent results that hold for all or certain subclasses of protocols. We state below several lemmas about message origination that are needed later to analyze concrete protocols. All those lemmas and their proofs are given in Appendix 8.1 and have been proved in our Isabelle/HOL formalization [6].

The first lemma specifies a lower bound on the time between when an agent first uses a nonce and another agent later uses the same nonce. The lemma holds whenever the initial knowledge of all agents does not contain any nonces.

Lemma 3.1. *Let A be an arbitrary (honest or dishonest) agent and let $(t^S_A, \mathit{Send}\ \mathit{Tx}^i_A\ \ m_A)$ be the first event in the trace tr containing the nonce N. If there is another event*

$(t_B^S, Send\ Tx_B^j\ m_B) \in tr$ with $A \neq B$ such that m_B contains N, then $t_B^S - t_A^S \geq cdist_{LoS}(A,B)$.

The next lemma is similar, but concerns the earliest possible time when an agent can receive a nonce.

Lemma 3.2. *Let A be an agent and let $(t_A^S, Send\ Tx_A^i\ m_A)$ be the first event in the trace tr containing nonce N in m_A. If tr contains an event $(t_B^R, Recv\ Rx_B^j\ m_B)$ where m_B contains N, then $t_B^R - t_A^S \geq cdist_{LoS}(A,B)$ holds.*

Our final lemma concerns signatures and their creation time.

Lemma 3.3. *Let A be an honest agent and let $(t_B^S, Send\ Tx_B^i\ m_B) \in tr$ be an event in the trace tr where the message m_B contains a signature of A. Then there is a send event $(t_A^S, Send\ Tx_A^j\ m_A) \in tr$ with m_A containing the same signature and $t_B^S - t_A^S \geq cdist_{LoS}(A,B)$.*

This lemma only holds if the initial knowledge of all agents does not contain the signing keys of other agents or signatures created by other agents. Additionally we assume that protocol messages never contain signing keys of agents. We formalize such assumptions as predicates on protocols and the initial knowledge.

4. Applying the Model

In this section, we use our model to analyze the security properties of three protocols: authenticated ranging, ultrasonic distance-bounding, and TESLA broadcast authentication. Each protocol uses cryptographic primitives as well as physical characteristics of the communication technology, environment, or network topology, in order to provide security guarantees. Since the first two protocols estimate distance based on round-trip measurements and bounds on the propagation speed of signals, variable clock offsets can trivially lead to wrong results. Therefore, we only consider those *ctime* functions that model a constant clock error. In the third example, we allow for arbitrary clock errors.

For the sake of convenience, we have chosen a simpler representation of the inductive rules defining the set of possible traces for each of the protocols below. In contrast to the definition in Section 3, where each protocol is defined by a set of step functions parameterizing the PROTO rule, we use one inductive rule per step function in the examples below. However, the equivalence of these two definitions is proved for each protocol in our Isabelle/HOL formalization [6]. These equivalence proofs allow us to use the protocol independent results presented in Section 3.5. For the interested reader, we present the set of step functions for the authenticated ranging example and give the equivalence proof in Appendix 8.2.

4.1. Authenticated Ranging

To define the set of possible traces for the authenticated ranging protocol introduced in Section 2.1, we give three rules modeling the agents' actions when executing the protocol:

1) The start rule (AR1) allows an agent to initiate a protocol run. We use r as the index of the radio transmitters and receivers of honest agents.

$$\frac{tr \in TR_{AR} \quad t_A^S \geq maxtime(tr) \qquad N_A \notin used(tr)}{tr.(t_A^S, Send\ Tx_A^r\ N_A) \in TR_{AR}}\ \text{AR1}$$

2) The reply rule (AR2) states that agents receiving an initial message may respond accordingly.

$$\frac{tr \in TR_{AR} \quad t_B^S \geq maxtime(tr) \qquad (t_B^R, Recv\ Rx_B^r\ N_A) \in tr}{tr.(t_B^S, Send\ Tx_B^r\ \{N_A, t_B^S - t_B^R\}_{SK_B}) \in TR_{AR}}\ \text{AR2}$$

3) The final rule (AR3) introduces a *Claim*-event. It models the conclusion of an initiator A who has received a response to his initial challenge.

$$\frac{tr \in TR_{AR} \quad t_A^R \geq maxtime(tr) \qquad (t_A^S, Send\ Tx_A^r\ N_A) \in tr \qquad (t_A^R, Recv\ Rx_A^r\ \{N_A, \delta\}_{SK_B}) \in tr}{tr.(t_A^R, Claim\ A\ (B, (t_A^R - t_A^S - \delta) * \frac{c}{2})) \in TR_{AR}}\ \text{AR3}$$

The premises state that A has initiated a protocol run and received a response from agent B. A therefore believes (as stated in the rule's conclusion) that $(t_A^R - t_A^S - \delta) * \frac{c}{2}$ is a reliable upper bound on the distance to B.

In combination with the rules NIL, FAKE, and NET, the rules AR1, AR2, and AR3 define all possible execution traces of the protocol. For this protocol, we define the initial knowledge of each agent A to be his own private key SK_A and the public keys PK_B of all agents B.

Security Analysis. As explained in Section 2.1, the protocol should compute a reliable upper bound on the physical distance between honest agents executing the protocol. We therefore state the following theorem:

Theorem 4.1. *Let A and B be honest agents, tr a valid trace, and $(t, Claim\ A\ (B,d)) \in tr$. Then $d \geq |loc_A - loc_B|$.*

For our proof we use the three protocol-independent lemmas about message ordering from Section 3, and the fact that for an honest agent B, δ sent in the second protocol message ($\delta = t_B^S - t_B^R$) always corresponds to the correct time interval between the *Recv*-event and *Send*-event (which follows directly from rule *AR*2).

Proof: Since only the rule AR3 adds events of the form $(t_A^C, Claim\ A\ \ (B,d))$, we know from the premises of AR3 that N_A originates at the event $(t_A^S, Send\ Tx_A^r\ N_A)$ in the trace.

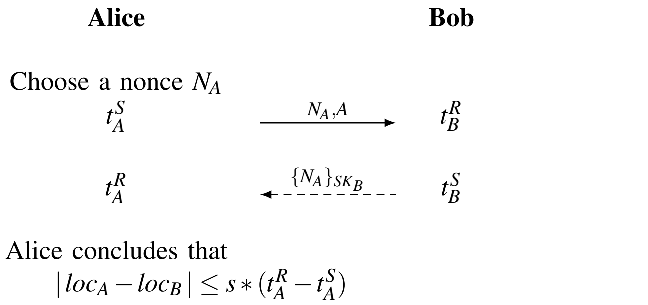

Figure 4: Distance Bounding Protocol (dashed arrow denotes ultrasound transmissions with speed s)

Furthermore, there is an event $(t_A^R, Recv\ Rx_A^r\ \ \{N_A, \delta\}_{SK_A})$, where $d = \frac{c}{2} * (t_A^R - t_A^S - \delta)$.
From the above, there must be a corresponding *Send*-event in the trace at time t_C^S, with $t_A^R - t_C^S \geq cdist_{LoS}(C, A)$, produced by some agent (possibly an intruder). Using Lemma 3.3, we conclude that B sent a message containing the signature at time t_B^S, where $t_C^S - t_B^S \geq cdist_{LoS}(B, C)$. This message must result from an application of rule AR2, since B is assumed to be honest. Hence there is a *Recv*-event at time t_B^R and $\delta = t_B^S - t_B^R$. Finally we use Lemma 3.2 to show that $t_B^R - t_A^S \geq cdist_{LoS}(A, B)$ and sum up the inequalities.

$$\begin{aligned} & t_A^R - t_A^S - \delta \\ = \quad & t_A^R - t_C^S + t_C^S - t_B^S + t_B^R - t_A^S \\ \geq \quad & cdist_{LoS}(C, A) + cdist_{LoS}(B, C) + cdist_{LoS}(A, B) \\ \geq \quad & 2 * cdist_{LoS}(A, B) \end{aligned}$$

Therefore we conclude that

$$\begin{aligned} d &= \frac{c}{2} * (t_A^R - t_A^S - \delta) \\ &\geq c * cdist_{LoS}(A, B) = |\,loc_A - loc_B\,|. \qquad \square \end{aligned}$$

4.2. Ultrasound Distance Bounding

Our second example is a protocol for *distance bounding* using ultrasound. The goal of the protocol in Figure 4 is for the initiator Alice to determine a reliable upper bound on the distance to a possibly dishonest responder Bob. Alice sends an unpredictable challenge N_A using radio signals and waits for the corresponding response on her ultrasound receiver. Then she measures the round-trip time and computes an upper bound $s * (t_A^R - t_A^S)$ on the distance. Using ultrasound (which is several orders of magnitude slower than radio), she can safely neglect the transmission time of the first message and the time needed for signing the response. Furthermore, using ultrasound allows the protocol to be implemented on off-the-shelf devices because time measurements with nanosecond precision are not required.

We assume that all agents A are equipped with ultrasound receivers Rx_A^{us} and transmitters Tx_A^{us}. Additionally every agent has a radio transmitter and receiver, Tx_A^r and Rx_A^r. If an ultrasound receiver Rx_B^{us} is able to receive messages from a transmitter Tx_A^i, then the communication distance should reflect that the message cannot be transmitted faster than s. We add the following properties of $cdist_{Net}$ as local assumptions for the security proof.

$$\begin{aligned} & cdist_{Net}(Tx_A^i, Rx_B^{us}) \neq \bot \\ & \Rightarrow cdist_{Net}(Tx_A^i, Rx_B^{us}) \geq \frac{|\,loc_A - loc_B\,|}{s} \end{aligned}$$

The same applies to messages transmitted by ultrasound transmitters Tx_A^{us} and received by receivers Rx_B^j.

$$\begin{aligned} & cdist_{Net}(Tx_A^{us}, Rx_B^j) \neq \bot \\ & \Rightarrow cdist_{Net}(Tx_A^{us}, Rx_B^j) \geq \frac{|\,loc_A - loc_B\,|}{s} \end{aligned}$$

We now give the inductive rules for the protocol. As in Section 4.1, we do not give the step functions, but present the equivalent definition that extends the protocol independent rules.

1) The start rule (DB1) initiates a protocol run.

$$\frac{tr \in TR_{DB} \quad t_A^S \geq maxtime(tr) \qquad N_A \notin used(tr)}{tr.(t_A^S, Send\ Tx_A^r\ N_A) \in TR_{DB}}\ \text{DB1}$$

2) The reply rule (DB2) allows receivers of initial messages to respond following the protocol.

$$\frac{tr \in TR_{DB} \quad t_B^S \geq maxtime(tr) \qquad (t_B^R, Recv\ Rx_B^r\ N_A) \in tr}{tr.(t_B^S, Send\ Tx_B^{us}\ \{N_A\}_{SK_B}) \in TR_{DB}}\ \text{DB2}$$

3) The final rule (DB3) introduces a *Claim*-event when an initiator A receives a response to his initial challenge.

$$\frac{tr \in TR_{DB} \quad t_A^R \geq maxtime(tr) \qquad (t_A^S, Send\ Tx_A^r\ N_A) \in tr \qquad (t_A^R, Recv\ Tx_A^{us}\ \{N_A\}_{SK_B}) \in tr}{tr.(t_A^R, Claim\, A\ (B, (t_A^R - t_A^S) * s)) \in TR_{DB}}\ \text{DB3}$$

This models what A concludes about a signal that apparently traveled back-and-forth between A and B in the time $t_A^R - t_A^S$, namely that $s * (t_A^R - t_A^S)$ is a reliable upper bound on their mutual distance.

Security Analysis. The security property of the distance bounding protocol is similar to the authenticated ranging example. But since the prover's computation time is not used in computing the distance, the protocol does not require an honest prover. We would expect a statement like the following to hold:

Proposition 4.2. *Let A be an honest agent and B be any agent. Furthermore consider a valid trace tr, where $(t, Claim\, A\ (B, d)) \in tr$. Then $d \geq |\,loc_A - loc_B\,|$.*

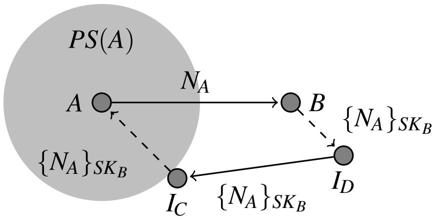

Figure 5: Attack on DB using Ultrasound

However this proposition is false without any further assumptions, as the attack in Figure 5 involving two colluding intruders shows. $PS(A)$ denotes the private space of A and is defined as the largest circle centered at A where A can ensure that no intruder is inside. To mount the attack, I_D is placed close to B, receiving B's reply over ultrasound. I_D forwards it to the second intruder I_C, close to A, using a radio link. I_C finally delivers the message to A using ultrasound. We have proven in Isabelle/HOL that this attack is captured in our model. The inequality involving the communication distances necessary for such an attack to work is

$$\begin{aligned}&cdist_{Net}(Tx_A^r, Rx_B^r) + cdist_{Net}(Tx_B^{us}, Rx_{I_D}^{us}) +\\ &cdist_{Net}(Tx_{I_D}^r, Rx_{I_C}^r) + cdist_{Net}(Tx_{I_C}^{us}, Rx_A^{us})\\ &< |\, loc_A - loc_B \,|/s\,.\end{aligned}$$

If the inequality holds, the intruders can speed up ultrasound communication between A and B (using their radio link) so that the deduced distance is smaller than the real distance between A and B.

In light of the above, we prove Proposition 4.2 under an additional assumption: The verifier A can ensure that the prover B is in his private space. The same assumption is used in other protocols (e.g., [12], [36]) for location-based access control and device pairing. This assumption is captured by the inequality

$$\forall I.\, |\, loc_A - loc_I | \;\geq\; |\, loc_A - loc_B \,|\,.$$

Note that this assumption thwarts Terrorist attacks (as defined in [9]), which would also falsify Proposition 4.2. We now restate Proposition 4.2, adding this additional assumption and proving the result.

Theorem 4.3. *Let A be an honest agent and B be an honest agent such that $\forall I.\, |\, loc_A - loc_I| \geq |\, loc_A - loc_B \,|$. Furthermore consider a valid trace tr, where $(t, Claim\, A\ (B,d)) \in tr$. Then $d \geq |\, loc_A - loc_B \,|$.*

Proof: The proof is by induction over traces and uses Lemma 3.1. Since only DB3 creates events of the form $(t_A^C, Claim\, A\ \ (B,d))$, we need not consider the other rules. From the premises of DB3, we conclude that the nonce N_A originates at the event $(t_A^S, Send\ Tx_A^r\ N_A)$. Furthermore, there must be an event $(t_A^R, Recv\ Rx_A^{us}\ \{N_A\}_{SK_A})$, where $d = s * (t_A^R - t_A^S)$. Therefore we must show that $t_A^R - t_A^S \geq |\, loc_A - loc_B \,|/s$.

Since there must be a *Send*-event corresponding to the *Recv*-event with the signature of B, the sender is either B or an intruder I. In the first case, the *Send* occurs at time t_B^S, with $t_A^R - t_B^S \geq cdist_{Net}(Tx_B^{us}, Rx_A^{us})$. From Lemma 3.1 it follows that $t_B^S \geq t_A^S$, since N_A is included in the message. Together with the previous inequality and the assumption that messages received by ultrasound receivers do not travel faster than s, we conclude that $t_A^R - t_A^S \geq cdist_{Net}(Tx_B^{us}, Rx_A^{us}) \geq |\, loc_A - loc_B \,|/s$.

In the second case, the message is sent by the intruder I at time t_I^S. Using the assumption that B is in the private space of A, it follows that the distance between A and I is at least the distance between A and B. Additionally, the assumptions state that a message received by Rx_A^{us} has not travelled with a speed faster than s. Together with $t_I^S \geq t_A^S$ (which follows from Lemma 3.1) this completes the proof.

$$\begin{aligned} t_A^R - t_A^S &\geq t_A^R - t_A^I\\ &\geq cdist_{Net}(Tx_I^j, Rx_A^{us})\\ &\geq |\, loc_A - loc_I \,|/s\\ &\geq |\, loc_A - loc_B \,|/s \quad \square\end{aligned}$$

Note that the proof does not use the fact that the second protocol message is authenticated by B. Correctness is guaranteed by A ensuring that B is in his private space. Therefore even a simplified version of the protocol, where the second message is replaced with the pair (N_A, B), would be secure under the private-space assumption.

4.3. A Delayed Key Disclosure Protocol

In our final example, we model and verify security properties of a *Delayed Key Disclosure* protocol used for broadcast authentication in resource constrained environments (such as sensor networks), where asymmetric cryptography is not available. In this type of protocol, the sender initially commits to a set of keys. To authenticate a message, he creates a keyed MAC using one of the (yet unpublished) keys to which he has committed. After all intended recipients have received the MAC, the sender opens the key commitment and therefore proves the origin of the message.

A suite of such protocols is described in [32]. We formalize the TESLA broadcast authentication protocol developed by Perrig et al. [31]. In TESLA, the sender commits to a set of keys $(K_i)_{1 \leq i \leq n}$, which are elements of a hash chain starting with a secret $H_0 (= K_n)$. The sender commits to them by publishing the hash-chain's last element H_n in an authentic way. Therefore every hash-chain element can be identified as such, by applying the hash function iteratively up to the point where the published element is reached. The one-way property of the hash function prevents the generation of elements prior to their release. The sender also publishes a key-release schedule that assigns keys to time intervals (validity windows) of length *valwin* and defines a

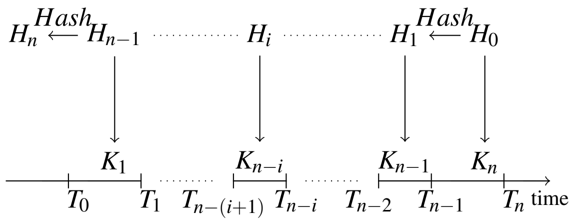

Figure 6: Association of Hash Chain Elements to Time-Slots

starting time T_0. Key $K_i = Hash^{n-i}(H_0)$ is then used within its validity window $[T_{i-1}, T_i[$, where $T_i = T_0 + i * valwin$, to generate a MAC for the messages sent in the same window. After K_i's validity window has passed, the sender releases the key K_i corresponding to the release schedule. We use $[T_{i+1}, T_{i+2}[$ as K_i's release window, which corresponds to the release schedule of TESLA defined in [31]. Figure 6 depicts the above.

In our formalization, we abbreviate $MAC_{K_i}(m) = (m, Hash(m, K_i))$ for the keyed MAC containing the message m. The secret H_0 is only contained in the broadcaster Br's initial knowledge and the initial knowledge of the other agents only contains H_n.

The protocol rules are formalized as follows.

1) The DKD1 rule formalizes the behavior of the broadcast source. According to the release schedule, Br chooses the currently valid key K_i and authenticates the message m. Additionally Br releases the old key K_{i-2}, valid in the interval $[T_{i-3}, T_{i-2}[$.

$$\dfrac{\begin{array}{c} tr \in TR_{DKD} \quad t \geq maxtime(tr) \\ t \in [T_{i-1}, T_i[\end{array}}{tr.(t, Send\ Tx_{Br}\ (MAC_{K_i}(m), K_{i-2})) \in TR_{DKD}}\ \text{DKD1}$$

2) The DKD2 rule models the conclusion of an agent R who received a message m authenticated with the key K_i before its expiration at T_i, according to the release schedule. In addition, the agent has received K_i at a later point in time.

$$\dfrac{\begin{array}{c} tr \in TR_{DKD} \quad t \geq maxtime(tr) \\ t^{R1} < T_i \quad i \leq n \\ (t^{R1}, Recv\ Rx_R\ (MAC_{K_i}(m), K_{i-2})) \in tr \\ (t^{R2}, Recv\ Rx_R\ (MAC_{K_{i+2}}(m'), K_i)) \in tr \end{array}}{tr.(t, Claim\ R\ (m,i)) \in TR_{DKD}}\ \text{DKD2}$$

Note that in premises of DKD2 we do not restrict the arrival time (t^{R2}) of the released key; it must just have arrived sometime. The premises could be further weakened by requiring only the reception of a later key (K_j, where $j > i$) allowing verification of all earlier keys even if the messages disclosing these have been lost.

Security Analysis. A broadcast protocol achieves *T-authentication* [37] if the protocol guarantees message-origin authentication, in combination with the guarantee that a received message has been sent by the claimed source within T time units before reception. We prove that TESLA achieves T-authentication for $T = valwin$.

Theorem 4.4. *Let tr be a valid trace. If $(t^C, Claim\ H_R\ (m,i)) \in tr$, then there exists a $(t^S, Send\ Tx_{Br}\ (MAC_{K_i}(m), K_{i-2})) \in tr$, where $t^S \in [T_{i-1}, T_i[$ holds.*

For simplicity of presentation, the presented proof assumes synchronized clocks. However, in our Isabelle/HOL formalization, we have proved that *valwin* is an upper bound on the clock error that is necessary and sufficient for the authentication property to hold. We prove Theorem 4.4 using two lemmas about the temporal secrecy of hash-chain elements.

The first lemma states that no other agent can use a key before it has been released by the broadcast source.

Lemma 4.5. *Suppose that $0 \leq l \leq n$, A is an agent other than Br, and tr is a valid trace. If $K_l \sqsubseteq DM_A(knows_A(tr))$ (i.e., the agent A can derive a message from his observations of the trace tr that contains K_l) or if $(t, Send\ Tx_A\ X) \in tr$, where $K_l \sqsubseteq X$, then $maxtime(tr) \geq T_{l+1}$.*

Proof: We prove this by induction on traces. The proof for the NIL and DKD2 rules follows from the induction hypothesis since these rules do not add any *Send*-events or *Recv*-events that change $DM_A(knows_A(tr))$. We now consider the three remaining rules.

FAKE**:** The event $(t_I, Send\ Tx_I^k\ X)$ is added to the trace tr. We must only consider the case where $K_l \sqsubseteq X$. Here, $K_l \sqsubseteq DM_A(knows_A(tr))$ follows from the premises of the rule and therefore $maxtime(tr) \geq T_{l+1}$ from the induction hypothesis.

CON**:** The event $(t_R, Recv\ Rx_A^k\ X)$ is added to the trace tr. We must only consider the case where a message containing K_l is added to some agent's knowledge. Hence $K_l \sqsubseteq X$ and there is a *Send*-event for X in tr as required by the premises of CON. The induction hypothesis can now be applied.

DKD1**:** The event $(t, Send\ Tx_{Br}\ (MAC_{K_i}(m), K_{i-2}))$ is added to the trace tr. Note that $K_{i-2} \sqsubseteq (MAC_{K_i}(m), K_{i-2})$, but K_i is not a part of the message since only the hash of K_i is included. $maxtime(tr) \geq t_{i+1}$ follows from the premises of the rule. □

In the next lemma, we claim that messages including $Hash(K_l, m)$, where the key K_l has not yet been released, must originate at the broadcaster.

Lemma 4.6. *Suppose that tr is a valid trace and that $(t, Send\ Tx_A\ M) \in tr$, where $Hash(K_l, m) \sqsubseteq M$. Furthermore suppose that $maxtime(tr) < T_{l+1}$, and $0 < l < n$. Then there exists an event $(\tilde{t}, Send\ Tx_{Br}\ (MAC_{K_l}(m), K_{l-2})) \in tr$, with $\tilde{t} \in [T_{l-1}, T_l[$.*

Proof: We must just consider the FAKE rule and the case where the event $(t_I, Send\ Tx_I^k\ X)$, with $Hash(K_l, m) \sqsubseteq X$, is added to the trace tr with $maxtime(tr) < T_{l+1}$.

$Hash(K_l,m) \sqsubseteq DM_I(knows_I(tr))$ follows from the rule's premises. This implies that either I received a message containing K_j for some $j \geq l$ or I received a message containing $Hash(K_l,m)$. But the first case is impossible since by Lemma 4.5, $maxtime(tr) \geq T_{l+1}$, which contradicts $maxtime(tr) < T_{l+1}$. The second case follows from the induction hypothesis since there must be a *Send*-event corresponding to the *Recv*-event in tr. □

The proof of Theorem 4.4 using the previous lemma is straightforward.

Proof: Since only DKD2 adds events of the form $(t, Claim\, H_R\, (m,i))$, we need not consider the other rules. From the premises of DKD2, we conclude that there is a *Recv*-event with message $(MAC_{K_i}(m), K_{i-2})$ and time t^{R1}, where $t^{R1} \in [T_{i-1}, T_i[$. Therefore, there must be a corresponding *Send*-event *sev* for the message, with $t^S < T_i$. We now consider the prefix of the trace up to *sev*. Since *sev* is the last event in the trace, $maxtime(tr) < T_{i+1}$ holds and using the premises from DKD2, we can apply Lemma 4.6, which completes the proof. □

In summary, the use of a theory combining cryptographic properties of messages with timed communication enabled us to verify TESLA, which uses time and properties of hash functions in a nontrivial way to achieve broadcast authentication.

5. Isabelle/HOL Formalization

We briefly survey our Isabelle/HOL formalization, providing additional details in Appendix 8.3. Our model builds on the following theories, depicted in Figure 7 along with their dependencies.

Message Theory: Our message theory (Section 3.3) models a free term algebra and is based on Paulson's work [30]. It also includes a formalization of hash chains and their properties.
Geometric Properties of $\mathbb{R}^3$*:* Since agents' locations are vectors in $\mathbb{R}^3$ (Section 3.2), we use the formalization of real numbers provided in Isabelle's standard library. Additionally we use the formalization of the Cauchy-Schwarz inequality [33] to establish relevant results about distances, like the triangle inequality.
Parameterized Communication Systems: Rules (Section 3.4) describe the network properties, possible intruder actions, and the protocol steps. Together these inductively define the set of possible traces.
Protocol Independent Properties: Parameterizing the set of possible traces by a protocol step function allows us to prove protocol independent system properties as described in Section 3.5.
Protocol Formalizations: These are given by sets of step functions (Section 4 and Appendix 8.2), formalizing the actions taken by agents running the protocol. For a given protocol, we instantiate the set of inductive rules with the corresponding step functions to obtain all possible execution traces. Security properties of the protocol are then proved by induction using the inherited protocol-independent facts.

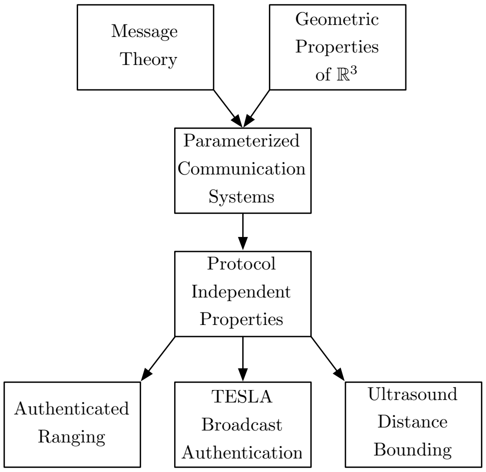
Message Theory
Geometric Properties of $\mathbb{R}^3$
Parameterized Communication Systems
Protocol Independent Properties
Authenticated Ranging
TESLA Broadcast Authentication
Ultrasound Distance Bounding

Figure 7: Dependency Graph of our Isabelle Theory Files

Most of our formalization consists of general results applicable to arbitrary protocols. The security proofs of the concrete protocols are therefore comparably small. Using Isabelle's support for structured proofs (Isar) results in proof scripts close to the proofs presented in this paper. Our complete formalization comprises 136 pages of PDF documentation (7103 lines) and our three examples take 11 pages (475 lines), 13 pages (579 lines), and 17 pages (857 lines) respectively.

Recently, we have used our framework to analyze a distance bounding protocol proposed by Meadows et al. in [26]. Despite the protocol's complexity, the resulting formalization and proofs (included in [6]) take only 13 pages (606 lines). Again, this is due to the reusable infrastructure provided by the framework presented in this paper.

6. Related Work

The formal analysis of security protocols is a very active research area. The two most popular approaches are based on automated methods, such as model checking [5], and interactive methods, such as theorem proving [30]. In both settings, it is standard to formalize an intruder model based on the Dolev-Yao model, which identifies the intruder with the network.

We now summarize formal approaches that address aspects of time, network topology, and location, which are the three central notions captured by our model. Most approaches formalizing time only focus on time-stamps, which are used to reason about key-expiration (e.g., in protocols like Kerberos). The models of [8], [15], [18] are based on discrete time, whereas [38] uses dense time. Corin et al. use timed automata [2] to model timing attacks and timing issues like timeouts and retransmissions in security protocols [14].

In [20] the authors use a real-time process algebra to model and analyze μ-TESLA. The protocol is proved to achieve a time-dependent form of integrity for a set of messages sent by the broadcast source, abstracting away from the network and the topology. Archer uses TAME [4] (an interface to PVS) in [3] to prove the authenticity of messages received in the correct validity window of the corresonding key in TESLA. In [22] Hopcroft and Lowe model two TESLA variants in CSP. Their formalization leads to a finite state space allowing for the automatic verification of the results from [3] using the model checker FDR.

Network topology has been considered in formal approaches for analyzing routing protocols in ad hoc networks [1], [27], [41]. Closely related is the notion of secure neighbor discovery (see for example [29]). In this setting, a node must detect its direct communication partners, for example, as a basis for topology information used for routing. In our model, a network's connectivity graph is described by the communication matrix and our formalization accounts for the difference between physical and communication distance. Whereas the goal of protocols like distance bounding is to establish a reliable upper bound on the Euclidean distance of two nodes, the output of a protocol seeking to guarantee secure neighborhood discovery should comply with the entries of the communication matrix. This observation shows that our model allows one to formally describe wormhole attacks and associated prevention mechanisms, e.g., as described in [23]. Furthermore, our model would be suitable for formal proofs of the impossibility results presented in [34].

Node location has been, to our knowledge, only used in informal proofs. For example, Sastry et al. [36] propose a protocol for verifying location claims based on ultrasonic communication and provide an informal proof of its security and reliability. Other approaches only formalize the related notion of relative distance. In Meadows et al. [26], an authentication logic is extended to handle relative distance and is used to prove the security of a newly proposed distance bounding protocol. Here, the distance between two nodes is axiomatically defined as the minimal time-of-flight of a message from the verifier to the prover and back. Different signal propagation speeds are not captured in the model.

7. Conclusion

We have presented a formal approach to modeling and verifying physical properties of security protocols for wireless networks. Our model captures dense time, agent locations, and physical properties of the communication network. To our knowledge, this is the first formal model that captures these aspects. This model has enabled us to formalize protocols, security properties, and environmental assumptions that are not amenable to formal analysis using other existing approaches. We have used our model to verify security properties of three different protocols and showed that our model captures relay attacks by distributed intruders.

As future work, we plan to extend our model to capture additional properties of wireless security protocols. We also intend to refine our model to capture message sizes and transmission rate, rapid bit exchange, and online guessing attacks.

References

[1] G. Acs, L. Buttyan, and I. Vajda, "Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1533–1546, 2006.

[2] R. Alur and D. Dill, "A Theory of Timed Automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[3] M. Archer, "Proving correctness of the basic TESLA multicast stream authentication protocol with TAME," in *Workshop on Issues in the Theory of Security*, 2002, pp. 14–15.

[4] M. Archer, C. Heitmeyer, and S. Sims, "TAME: A PVS interface to simplify proofs for automata models," in *Proceedings of User Interfaces for Theorem Provers*, 1998.

[5] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago *et al.*, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," *CAV '05: Proceedings of the 17th Conference on Computer Aided Verification*, 2005.

[6] B. Schmidt and P. Schaller, "Isabelle Theory Files: Modeling and Verifying Physical Properties of Security Protocols for Wireless Networks," http://people.inf.ethz.ch/benschmi/ProtoVeriPhy/.

[7] C. Ballarin, "Interpretation of locales in Isabelle: Theories and proof contexts," *Mathematical Knowledge Management (MKM 2006), LNAI*, vol. 4108, 2006.

[8] G. Bella, *Formal Correctness of Security Protocols (Information Security and Cryptography)*. Springer-Verlag New York, Inc., 2007.

[9] S. Brands and D. Chaum, "Distance-bounding protocols," in *EUROCRYPT '93: Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*. Springer-Verlag New York, Inc., 1994, pp. 344–359.

[10] S. Capkun and J. Hubaux, "Secure positioning in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 221–232, 2006.

[11] S. Capkun, L. Buttyan, and J.-P. Hubaux, "SECTOR: secure tracking of node encounters in multi-hop wireless networks," in *SASN '03: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*. New York, NY, USA: ACM Press, 2003, pp. 21–32.

[12] S. Capkun and M. Cagalj, "Integrity regions: authentication through presence in wireless networks," in *WiSe '06: Proceedings of the 5th ACM Workshop on Wireless Security*. New York, NY, USA: ACM Press, 2006, pp. 1–10.

[13] J. Clulow, G. P. Hancke, M. G. Kuhn, and T. Moore, "So near and yet so far: Distance-bounding attacks in wireless networks," in *Security and Privacy in Ad-hoc and Sensor Networks*. Springer, 2006, pp. 83–97.

[14] R. Corin, S. Etalle, P. Hartel, and A. Mader, "Timed analysis of security protocols," *Journal of Computer Security*, vol. 15, no. 6, pp. 619–645, 2007.

[15] G. Delzanno and P. Ganty, "Automatic Verification of Time Sensitive Cryptographic Protocols," *TACAS '04: Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2004.

[16] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE, Transactions on Information Theory*, no. 2(29), pp. 198–208, 1983.

[17] S. Drimer and S. J. Murdoch, "Keep your enemies close: distance bounding against smartcard relay attacks," in *Usenix '07: Proceedings of 16th USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2007, pp. 1–16.

[18] N. Evans and S. Schneider, "Analysing Time Dependent Security Properties in CSP Using PVS," in *ESORICS '00: Proceedings of the 6th European Symposium on Research in Computer Security*. London, UK: Springer-Verlag, 2000, pp. 222–237.

[19] S. Ganeriwal, S. Capkun, C.-C. Han, and M. B. Srivastava, "Secure time synchronization service for sensor networks," in *WiSe '05: Proceedings of the 4th ACM Workshop on Wireless Security*. ACM Press, 2005, pp. 97–106.

[20] R. Gorrieri, F. Martinelli, M. Petrocchi, and A. Vaccarelli, "Formal analysis of some timed security properties in wireless protocols," in *FMOODS '03: Proceedings of the 6th IFIP Workshop on Formal Methods for Open Object-based Distributed Systems*, 2003, pp. 139–154.

[21] G. P. Hancke and M. G. Kuhn, "An RFID distance bounding protocol," in *SECURECOMM '05: Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 67–73.

[22] P. J. Hopcroft and G. Lowe, "Analysing a stream authentication protocol using model checking," *International Journal of Information Security*, vol. 3, no. 1, pp. 2–13, 2004.

[23] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *INFOCOM '03*, vol. 3, 2003, pp. 1976–1986.

[24] M. Kuhn, "An Asymmetric Security Mechanism for Navigation Signals," *IH 2004: 6th International Workshop on Information Hiding, Revised Selected Papers*, 2004.

[25] L. Lazos, R. Poovendran, and S. Capkun, "ROPE: robust position estimation in wireless sensor networks," *IPSN 2005: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, 2005*, pp. 324–331, 2005.

[26] C. Meadows, R. Poovendran, D. Pavlovic, L. 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*, pp. 279–298, 2006.

[27] S. Nanz and C. Hankin, "A framework for security analysis of mobile wireless networks," *Theoretical Computer Science*, vol. 367, no. 1, pp. 203–227, 2006.

[28] T. Nipkow, L. Paulson, and M. Wenzel, *Isabelle/Hol: A Proof Assistant for Higher-Order Logic*. Springer, 2002.

[29] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun, and J. Hubaux, "Secure neighborhood discovery: a fundamental element for mobile ad hoc networking," *Communications Magazine, IEEE*, vol. 46, no. 2, pp. 132–139, 2008.

[30] L. C. Paulson, "The inductive approach to verifying cryptographic protocols," *Journal of Computer Security*, vol. 6, pp. 85–128, 1998.

[31] A. Perrig, R. Canetti, J. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," *RSA CryptoBytes*, vol. 5, no. 2, pp. 2–13, 2002.

[32] A. Perrig and J. D. Tygar, *Secure Broadcast Communication in Wired and Wireless Networks*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[33] B. Porter, "Cauchy's mean theorem and the cauchy-schwarz inequality," in *The Archive of Formal Proofs*, G. Klein, T. Nipkow, and L. Paulson, Eds. http://afp.sf.net/entries/Cauchy.shtml, Mar. 2006, formal proof development.

[34] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility," in *ASIACCS '08: Proceedings of the 3nd ACM Symposium on Information, Computer and Communications Security*. ACM, 2008, pp. 189–200.

[35] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, "Detecting relay attacks with timing-based protocols," in *ASIACCS '07: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*. ACM, 2007, pp. 204–213.

[36] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*. New York, NY, USA: ACM Press, 2003, pp. 1–10.

[37] P. Schaller, S. Capkun, and D. Basin, "BAP: Broadcast authentication using cryptographic puzzles," *ACNS 2007: International Conference on Applied Cryptography and Network Security*, vol. 4521, pp. 401–419, 2007.

[38] R. Sharp and M. Hansen, "Timed Traces and Strand Spaces," *Proceedings of the 16th Nordic Workshop on Programming Theory*, pp. 96–98, 2004.

[39] V. Shmatikov and M. Wang, "Secure Verification of Location Claims with Simultaneous Distance Modification," *Lecture Notes in Computer Science*, vol. 4846, p. 181, 2007.

[40] K. Sun, P. Ning, and C. Wang, “TinySeRSync: secure and resilient time synchronization in wireless sensor networks,” *CCS ’06: Proceedings of the 13th ACM conference on Computer and Communications Security*, pp. 264–277, 2006.

[41] S. Yang and J. S. Baras, “Modeling vulnerabilities of ad hoc routing protocols,” in *SASN ’03: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*. New York, NY, USA: ACM, 2003, pp. 12–20.

8. Appendix

8.1. Proofs of the Protocol Independent Lemmas

Lemma 8.1. *Let A be an arbitrary (honest or dishonest) agent and let $(t_A^S, Send\ Tx_A^i\ m_A)$ be the first event in the trace tr containing the nonce N. If there is another event $(t_B^S, Send\ Tx_B^j\ m_B) \in tr$ with $A \neq B$ such that m_B contains N, then $t_B^S - t_A^S \geq cdist_{LoS}(A,B)$.*

Proof: We prove this by induction on traces. The proof for the rules NIL and NET follows trivially from the induction hypothesis since these rules do not add *Send*-events. We now consider the two remaining rules. Let A be an agent and $(t_A^S, Send\ Tx_A^i\ m_A)$ the first event containing the nonce N.
FAKE**:** An event $(t_I^S, Send(Tx_I^k, m_I))$ is appended to the trace. The only interesting cases are the ones where $A \neq I$ and m_I contains N. From the assumptions of the rule, we have $m_I \in DM_I(knows_I(tr))$. Since I cannot synthesize a nonce created by A, I must have received a message containing N at time t_I^R, where $t_I^R \leq t_I^S$. Since every *Recv*-event is preceded by a corresponding *Send*-event, there must be such an event in the trace occurring at some agent C at time t_C^S, where $t_C^S \leq t_I^R - cdist_{Net}(Tx_C^l, Rx_I^h)$. From the induction hypothesis, we have $t_C^S - t_A^S \geq cdist_{LoS}(A,C)$. Using the triangle inequality for the physical distance and the consistency condition forbidding faster-than-light communication, $t_I^S - t_A^S \geq cdist_{LoS}(A,I)$ immediately follows.
PROTO**:** The event $(t_B^S, translateEv(B, action, m_B))$ is appended to the trace. Only the case where $pevType = SendA$, $A \neq B$, and m_B contains N is interesting. From the assumptions of the rule, we have $m_B \in DM_B(knows_B(tr))$, like in FAKE. The rest of the proof is analogous to the FAKE case since the same network and message derivation rules apply to honest and dishonest nodes. □

Lemma 8.2. *Let A be any agent and let $(t_A^S, Send\ Tx_A^i\ m_A)$ be the first event in a trace tr containing the nonce N in m_A. If tr contains an event $(t_B^R, Recv\ Rx_B^j\ m_B)$ where m_B contains N, then $t_B^R - t_A^S \geq cdist_{LoS}(A,B)$ holds.*

Proof: We prove this by induction on traces considering the individual rules. Here, only the NET-rule is of interest since the proof that the lemma still holds after adding *Claim*-events and *Send*-events is trivial. The *Net*-rule adds a *Recv*-event $(t_B^R, Recv\ Rx_B^k\ m)$, which must always be preceded by a corresponding *Send*-event $(t_C^S, Send\ Tx_C^l\ m)$, where $t_C^S \leq t_B^R - cdist_{Net}(Tx_C^l, Rx_B^k)$. From Lemma 8.1, we know that $t_C^S - t_A^S \geq cdist_{LoS}(A,C)$ for the event that generates the nonce at time t_A^S. Combining the two inequalities and using the triangle inequality and the consistency condition for $cdist_{Net}$, we get the desired inequality $t_B^R - t_A^S \geq cdist_{LoS}(A,B)$. □

Lemma 8.3. *Let A be an honest agent and let $(t_B^S, Send\ Tx_B^i\ m_B) \in tr$ be an event in the trace tr where the message m_B contains a signature of A. Then there is a send event $(t_A^S, Send\ Tx_A^j\ m_A) \in tr$ where m_A contains the same signature and $t_B^S - t_A^S \geq cdist_{LoS}(A,B)$.*

Proof sketch: The proof is analogous to the proof of Lemma 8.1. The proof additionally uses the fact that intruders cannot create signatures on behalf of honest agents since the signing keys of honest agents are never leaked. □

8.2. Representing the Set of Possible Traces

In Section 4, we modeled each example protocol by extending the NIL, FAKE, and NET rules with a set of inductive rules describing the agents’ actions associated with the corresponding protocol. These protocol-specific rules act directly on a given trace, in contrast to our representation in Section 3, where each protocol is defined by a set of *step* functions, parameterizing the PROTO rule. The advantage of the *step* function approach is that the inductive rules are parameterized by the protocol, and we can therefore prove protocol-independent theorems, reusable in all protocol instantiations. However, we have presented the protocols in Section 4 using protocol-dependent inductive rules because of their more intuitive character.

In our Isabelle/HOL formalization [6], we have shown the equivalence of the definitions for all three protocols. In this section we give the definition of the authenticated ranging protocol analyzed in Section 4.1 as a set of *step* functions, and the proof of equivalence of both representations.

8.2.1. Step Functions for the AR Protocol.

As explained in Section 3, step functions are of type $agent \times trace \times real \rightarrow (action \times msg)\ set$. The step functions formalizing the behavior of the agents participating in an execution of the authenticated ranging protocol, as described in Figure 1, are the following:

Start: An agent A can start a protocol run by sending a fresh nonce NA at a local time t_A^S.

$$
\begin{aligned}
&ar1(A,tr,t) \equiv \\
&\qquad \bigcup_{NA} \{(\mathsf{SendA}\ \ r, Nonce\ A\ NA) \mid \\
&\qquad\qquad Nonce\ A\ NA \notin used(tr)\}
\end{aligned}
$$

Reply: If an agent B receives a nonce NA at time t_B^R, he may continue the protocol by replying with the message

$\{NA, \delta\}_{SK_B}$, where $\delta := t_B^S - t_B^R$ denotes the time difference, between sending this message and receiving the nonce NA.

$$
\begin{aligned}
&ar2(B, tr, t_B^S) \equiv \\
&\quad \bigcup_{N, t_B^R, C} \{(\mathsf{SendA}\ r, \{N, t_B^S - t_B^R\}_{SK_B}) \mid \\
&\qquad (t_B^R, Recv\ (Rx_B^r)\ N) \in tr\}
\end{aligned}
$$

Conclusion: Suppose the initiator of a protocol run receives back the initial nonce NA paired with a time interval δ, and both are signed with the signing key of an agent B (i.e., $\{NA, \delta\}_{SK_B}$) at time t_A^R. Then he concludes that $(t_A^R - t_A^S - \delta) * s$ is a reliable upper bound on the distance to agent B, where s denotes the speed of the communication medium.

$$
\begin{aligned}
&ar3(A, tr, t) \equiv \\
&\quad \bigcup_{NA, B, t_A^S, \delta, t_A^R} \{(\mathsf{ClaimA}\ , (Agent\ B, (t_A^R - t_A^S - \delta) * v)) \mid \\
&\qquad (t_A^R, Recv\ (Rx_A^r)\ \{NA, \delta\}_{SK_B}) \in tr \\
&\qquad (t_A^S, Send\ (Tr_A^r)\ NA) \in tr\}
\end{aligned}
$$

Therefore the set $ar = \{ar1, ar2, ar3\}$ is the set of *step* functions defining all possible steps of agents executing the authenticated ranging protocol given in Figure 1. In Section 4.1 we gave the rules $AR1$, $AR2$, and $AR3$. The equivalence between both definitions is stated by the following theorem:

Theorem 8.4. *The inductive set $Tr(ar)$ defined by the rules* NIL, FAKE, NET, *and* PROTO *is equal to the inductive set TR_{AR}, where the* PROTO *rule is replaced by the* AR1, AR2, *and* AR3 *rules, i.e. $Tr(ar) = TR_{AR}$.*

Proof: We prove the theorem by establishing inclusion in both directions.

$\subseteq$: We show that an arbitrary trace $tr \in Tr(ar)$ is an element of TR_{AR}. This is done by induction on the trace tr using the induction principle induced by $Tr(ar)$, i.e. we must show that for each rule in the definition of $Tr(ar)$, the extended trace is also in TR_{AR}. For the NIL, FAKE, and NET rules, this is trivial since the definitions coincide. For the PROTO rule we must consider the three cases for $step \in ar$ separately and show that each event added by a step function ari corresponds to an event appended to the trace by the corresponding ARi rule.

$\supseteq$: We consider an arbitrary $tr \in TR_{AR}$ and show that it is also in $Tr(ar)$. We apply the induction principle introduced by the definition of TR_{AR}. To show the inclusion for traces extended by the ARi rules, we apply the PROTO rule. To simplify the proof that the assumptions of the PROTO hold, we have shown in a separate step that $ari(A, view_A(tr), localtime_A(t)) = ari(A, tr, t)$ for all possible values of A, tr, and t, and that the returned message is derivable. This just means that the protocol is executable, i.e. each agent only considers local events for his decisions and can derive all outgoing messages. Additionally, we use the fact that the AR-protocols only relies on time differences, not on absolute times. Note that we assume constant clock offset for this example. □

The proofs for the remaining cases of the presented distance bounding protocol and the delayed key disclosure protocol are not presented here. However the proof technique is identical: we use induction to establish each inclusion. Proofs are given in our Isabelle/HOL formalization and can be found at [6].

8.3. Details on the Isabelle/HOL Formalization

The use of parameterized inductive definitions, as described in Section 5, is one technique that we have used to model a general theory and prove protocol-independent facts. The other technique is underspecification, supported in Isabelle/HOL by using uninterpreted constants. For example, we have declared *loc* as an uninterpreted function constant of type $agent \rightarrow \mathbb{R}^3$. The results then proved, involving this function, hold for all *loc* functions of the declared type.

Specifying properties of a function (beyond just its type) requires adding the properties as assumptions to all lemmas that use the function. Isabelle provides two constructs to simplify such bookkeeping.

The first construct is the **specification** construction that defines a function using the Hilbert-choice operator ε as $f \equiv \varepsilon f.P(f)$, i.e. f is some function such that the property $P(f)$ holds. To ensure that no inconsistencies are introduced, one must prove that at least one such function exists. We have, for example, used this specification mechanism to specify that $cdist_{Net}$ is never negative and always greater or equal than $cdist_{LoS}$.

The second construct is based on locales [7], which are Isabelle's mechanism to formalize parametric theories. Locales allow the definition of proof contexts where one may specify formulas that hold for all instantiations of these contexts. In our formalization, we use locales mainly to achieve parametricity in the initial state and the protocol run by honest agents. For example, the *INITKNOWS* context declares an *initKnows* function. Later on, we refine this context in various ways by adding different assumptions about the initial knowledge. Results like Lemma 3.1 about nonce origination are proven in such contexts. The initial knowledge for concrete protocols is then defined in the protocol-specific theory files. Interpretation of contexts allows us to reuse the parameterized proofs for the specific protocols. We also use locales to formalize classes of protocols and in proofs of protocol independent properties.