Formal Reasoning about Physical Properties of Security Protocols

DAVID BASIN, SRDJAN CAPKUN, PATRICK SCHALLER, and BENEDIKT SCHMIDT
ETH Zurich, Switzerland

Traditional security protocols are mainly concerned with authentication and key establishment and rely on predistributed keys and properties of cryptographic operators. In contrast, new application areas are emerging that establish and rely on properties of the physical world. Examples include protocols for secure localization, distance bounding, and secure time synchronization.

We present a formal model for modeling and reasoning about such physical 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 between nodes. 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 have used it to verify protocols for authenticated ranging, distance bounding, broadcast authentication based on delayed key disclosure, and time synchronization.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Protocol verification*

General Terms: Protocol verification, wireless networks

Additional Key Words and Phrases: Interactive theorem proving, formal models

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 employed protocols. 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 [Brands and Chaum 1994; Capkun et al. 2003; Hancke and Kuhn 2005; Meadows et al. 2006; Rasmussen and Capkun 2010].

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 [Capkun and Hubaux 2006; Kuhn 2004; Lazos et al. 2005; Sastry et al. 2003]. Secure localization and physical proximity verification protocols, and attacks on them, have been implemented on RFID, smart cards, and Ultra-Wide Band

Authors' addresses: Universitätstrasse 6, 8092 Zürich, Switzerland

platforms [Drimer and Murdoch 2007; Reid et al. 2007; Rasmussen and Capkun 2010].

Secure Time Synchronization. A node must securely synchronize its clock to the clock of another trusted node in an adversarial setting [Ganeriwal et al. 2008; Sun et al. 2006]. This kind of protocol also serves as a building block for other security protocols such as broadcast authentication [Perrig and Tygar 2002].

Secure Neighbor Discovery or Verification. A node must determine or verify its direct communication partners within a communication network [Papadimitratos et al. 2008]. Reliable information about the topology of a network 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.

In this article, we present a formal model for reasoning about 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 as 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 that they exchange 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 those of the classical Dolev-Yao intruder.

Our model combines a message and a communication model. Whereas cryptographic aspects of protocol messages are captured in the symbolic message model, our communication model allows us to express relevant properties of the communication technology. Similar to Paulson's *Inductive Approach* [Paulson 1998], we have used Isabelle/HOL [Nipkow et al. 2002] to formalize our model and to prove security properties of the protocols presented in this article. We model communication as traces of send and receive events, where the communication technology and the network topology determine the times and locations of the receive events resulting from a given send event.

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

2. CONTRIBUTION

Our model bridges the gap between informal approaches used to analyze physical protocols for wireless networks and the formal approaches taken for security protocol analysis. Informal approaches typically demonstrate the absence of a given set of attacks, rather than proving that the protocol works correctly in the pres-

ence of an active adversary taking arbitrary actions. In contrast, existing formal approaches fail to capture the details necessary to model physical protocols and their intended properties. To bridge this gap, our model formalizes an operational, trace-based semantics of security protocols that accounts for time, location, network topology, and distributed intruders. To model cryptographic operators and message derivability, we reuse parts of standard modeling approaches based on the perfect cryptography assumption.

In what follows, we explain our contributions in more detail. First, we give a novel operational semantics that captures the essential physical properties of space and time and thereby supports natural formalizations of many physical protocols and their corresponding security properties. For example, properties may be stated in terms of the relative distance between nodes, the locations of nodes, and the times associated with the occurrence of events. Moreover, protocols can compute with, and base decisions upon, these quantities.

Second, despite its expressiveness, our operational semantics is still simple and abstract enough to allow its complete formalization. We have formalized our model in Isabelle/HOL and used it to formally derive both protocol-independent and protocol-specific properties, directly from the semantics. Protocol-independent properties formalize properties of communication and cryptography, independent of any given protocol. For example, it follows from our operational semantics that there are no collisions for randomly chosen nonces and that communication cannot travel faster than the speed of light. This allows us to prove in a protocol-independent way a lower bound on the time until an adversary learns a nonce depending on his distance to the node generating the nonce. We use these properties, in turn, to prove protocols correct[1] or to uncover weaknesses or missing assumptions through unprovable subgoals.

Finally, we show that our approach is viable for the mechanized analysis of a range of wireless protocols. We demonstrate this by providing four case studies that highlight different features of the model:

(1) Our formalization of an authenticated ranging protocol shows how time-of-flight measurements of signals relate to physical distances between nodes. Additionally, the model has to account for local computation times which are included in protocol messages.

(2) Our formalization of an ultrasound distance bounding protocol demonstrates how the model accounts for transceivers that employ different communication technologies and their interaction. Furthermore, the example shows how our notion of location can be used to formalize private space assumptions.

(3) Our model of a protocol based on delayed key disclosure shows how to handle arbitrary clock offsets. The example also includes nontrivial cryptographic reasoning about hash chains and MACs.

(4) Our formalization of a secure time synchronization protocol illustrates how we can model relations between local clock offsets of different nodes. It also shows

[1]Technically speaking, we carry out our formalization in higher-order logic, conservatively extended by our operational semantics. All theorems about our model and specific protocols are proven as theorems in this conservative extension.

how bounds on the message transmission time can be specified.

The rest of the article is organized as follows. In Section 3, we present an example protocol and background information on protocol verification and Isabelle/HOL. In Section 4, we present our model. In Section 5, we describe the protocols that we formalize and the proofs of their security properties. Finally, we discuss our formalization, survey related work, and draw conclusions in in Sections 7, 6, and 8.

3. BACKGROUND

3.1 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 e.g., [Brands and Chaum 1994; Capkun and Hubaux 2006] for details on authenticated ranging and [Tippenhauer and Capkun 2009] for a description of its RF implementation. As an application of such a protocol, consider a door-locking system that requires that a legitimate key, such as an RFID card or smart key, must be within a given distance of the door for the lock to open. Verifying this distance is critical for the security of the application. For example, protocols for passive keyless entry and start systems that do not verify the proximity have recently been shown to be vulnerable to relay attacks [Francillion et al. 2010].

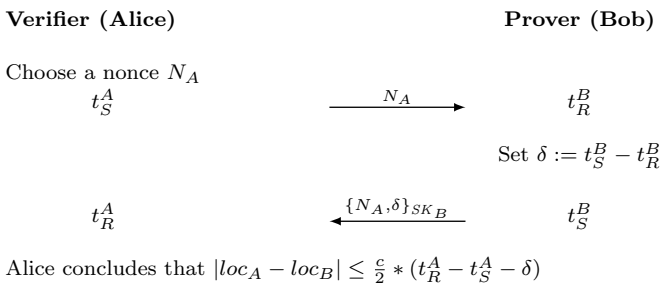


Fig. 1. Authenticated Ranging Protocol

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 messages. She uses the protocol to measure the round-trip time of a radio signal traveling at the speed of light c between her and Bob. In particular, she creates a fresh, unguessable nonce N_A and sends it to Bob at time t_S^A. 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. If Bob cannot predict the processing time, he can choose δ sufficiently large and then delay sending the message accordingly. Upon receiving the reply, Alice notes the time of reception t_R^A and calculates the time-of-flight, $t_R^A - t_S^A - \delta$. Since the computation time δ on the prover's side is subtracted in the calculation of the distance, the prover must be trusted. Otherwise

a malicious prover could report to the verifier a $\bar{\delta}$ that is greater than the actual processing time δ, thereby decreasing the computed round-trip time.

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

3.2 The Inductive Approach

In [Paulson 1998] Paulson has introduced an inductive approach to security protocol verification. The 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 events executed by the involved agents. In particular, protocols are modeled by rules describing the protocol steps executed by honest agents and possible intruder actions. The rules constitute 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 specified as predicates on traces. Protocol security is then proved by induction on traces using an induction principle derived from the protocol rules. Paulson formalized his approach within higher-order logic in the Isabelle/HOL system and used it to prove security properties for a wide range of security protocols.

The model we present in this article extends Paulson's inductive approach to allow us reasoning about security properties that involve physical aspects of the environment, such as the propagation delay and a notion of time. Like Paulson, we have formalized our approach in Isabelle/HOL and have reused parts of his formalization of the message model. However, we have refined the communication model to take physical aspects of the environment into account. In Section 3.3, we present Isabelle/HOL specific notation that we will use to present our model.

3.3 Isabelle/HOL

Isabelle [Nipkow et al. 2002] 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. We use the predefined list type $\alpha\ list$ where $xs.x$ denotes the list xs extended by the element x. Algebraic data types are 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 4.

4. FORMAL MODEL

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

4.1 Modeled Concepts

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 received messages.

Network. We model an unreliable network connecting agents' transmitters and receivers as a communication 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 there is a corresponding entry in the communication 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 different communication technologies with different signal propagation speeds, e.g., radio and ultrasound technologies. 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 the physical distance corresponds to the Euclidean distance, the network topology describes signal paths not necessarily corresponding to the line-of-sight paths between senders and receivers, such as rolled up cables or 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 the time required for line-of-sight transmission. Since these laws are universal, our model applies to any kind of network where the network topology can be described by a fixed commnunication matrix.

Time. Protocols, such as the authenticated ranging example from Section 3.1, measure time to make statements about distances. 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 time. We achieve this by tagging every event with a corresponding timestamp. We model temporal dependencies and clock access by agents using rules that account for arbitrary offsets of local clocks.

Intruder Model. In order to reason formally about properties of security protocols we must precisely define the capabilities of the intruder. We therefore need to specify the intruder's capabilities in terms of network control, as well as the intruder's cryptographic capabilities. The most prominent and most widely used intruder model is the so called Dolev-Yao intruder [Dolev and Yao 1983]. In this model, an intruder completely controls the communication network in the sense that he can overhear, remove, and delay any message sent by honest agents, as well as insert any message he is able to construct according to his cryptographic capabilities. In terms of cryptographic capabilities, the Dolev-Yao intruder implements the so called perfect cryptography assumption.

In our model, the intruder's cryptographic capabilities correspond to those of

the Dolev-Yao intruder. However, in terms of network control, our communication model is subject to physical restrictions, such as transmission time and network topology. These 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.

Note that these extensions are essential for modeling protocols that involve physical properties of the environment such as time and location. Such protocols fall outside the scope of standard symbolic protocol models based on the Dolev-Yao intruder. This is understandable: the Dolev-Yao model was developed for classical security protocols which do not rely on properties of the physical environment.

4.2 Agents and the Environment

We now present our model and sketch its formalization in Isabelle/HOL. Technical details of our formalization can be found in the Isabelle/HOL theory files [Schmidt and Schaller 2010].

Agents and Transmitters. Agents are either honest or intruders. We model infinitely many agents of each kind by using the set of natural numbers $\mathbb{N}$ as agent identifiers.

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

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.

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

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

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

Physical and Communication Distance. The function loc assigns to 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 $|\, loc_A - loc_B \,|$.

The line-of-sight distance between the locations of the agents A and B in $\mathbb{R}^3$ is the shortest path, taken for example by electromagnetic waves when there are no obstacles. We define the line-of-sight communication distance as the time it takes for a radio signal to travel this path,

$$\mathit{cdist}_{LoS}(A, B) = \frac{|\, loc_A - loc_B \,|}{c}\,.$$

The value computed by cdist_{LoS} only depends on A and B's location and is independent of the network topology. We model the network topology using the function

$cdist_{Net} : transmitter \times receiver \to \mathbb{R}_{\geq 0} \cup \{\bot\}$, whose value depends on the communication medium used by the given transceivers, obstacles between transmitters and receivers, and other environmental factors. $cdist_{Net}(Tx^i_A, Rx^j_B) = \bot$ denotes that Rx^j_B cannot receive transmissions from Tx^i_A. In contrast, $cdist_{Net}(Tx^i_A, Rx^j_B) = t$, where $t \neq \bot$, describes that Rx^j_B may receive messages emitted by Tx^i_A after a minimum delay of t time units. Since we assume that information cannot propagate faster than with the speed of light, we always require that

$$cdist_{LoS}(A, B) \leq cdist_{Net}(Tx^i_A, Rx^j_B).$$

In Isabelle/HOL, we model *loc* as an uninterpreted function. That is, we we give *loc* a type, but do not provide a concrete interpretation. Similarly, $cdist_{Net}$ is uninterpreted, but we restrict the class of possible interpretations by additionally requiring the previously mentioned property: faster-than-light communication is impossible. Further assumptions about the agents' locations and the network topology needed for analyzing protocols can be added as local assumptions in security proofs. As an example of such an additional assumption, consider the ultrasound distance bounding protocol and its security properties described in Section 5.2. For the protocol to have the expected security properties, we must assume that there is no adversary in a given area (the so called *private space*) around an honest agent. This is therefore modeled as an additional assumption in the corresponding security proof. Hence, our results apply to all possible locations of agents and to all network topologies that fulfill the corresponding 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 which correspond to 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.

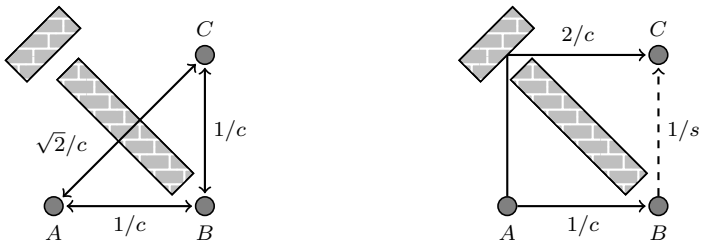


Fig. 2. Physical (left) and Network Topology (right).

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 diagonal wall in the middle prevents line-of-sight communication from A to C. However, reflections off the wall in the upper left corner enable 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.

4.3 Messages and Events

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

$$\begin{array}{llll} \textbf{datatype}\ msg = \mathsf{Agent}\ agent & \mid \mathsf{Time}\ \mathbb{R} & \mid \mathsf{Number}\ \mathbb{N} & \mid \mathsf{Nonce}\ agent\ \mathbb{N} \\ \quad \mid \mathsf{Key}\ key & \mid \mathsf{Hash}\ msg & \mid \mathsf{MPair}\ msg\ msg & \mid \mathsf{Crypt}\ key\ msg \end{array}$$

Nonces play the role of random, unguessable strings and are tagged with the name of the agent who created them and a unique identifier. To create a fresh nonce, the *used* predicate introduced below can be used to ensure that the nonce is fresh. Tagging nonces with the creator's name ensures that nonces created by different agents never collide. Indeed, even colluding intruders must communicate to share a nonce. Similar to nonces, keys are tagged with 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 $DM : agent \to msg\ set \to msg\ set$. 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.

$$\frac{m \in M}{m \in DM_A(M)}\ \text{INJ} \qquad \frac{m \in DM_A(M)}{Hash\ m \in DM_A(M)}\ \text{HASH} \qquad \frac{(m,n) \in DM_A(M)}{m \in DM_A(M)}\ \text{FST}$$

$$\frac{(m,n) \in DM_A(M)}{n \in DM_A(M)}\ \text{SND} \qquad \frac{m \in DM_A(M) \qquad n \in DM_A(M)}{(m,n) \in DM_A(M)}\ \text{PAIR}$$

$$\frac{m \in DM_A(M) \qquad Key\ k \in DM_A(M)}{\{m\}_k \in DM_A(M)}\ \text{ENC} \qquad \frac{}{Nonce\ A\ n \in DM_A(M)}\ \text{NONCE}$$

$$\frac{\{m\}_k \in DM_A(M) \qquad (Key\ k)^{-1} \in DM_A(M)}{m \in DM_A(M)}\ \text{DEC} \qquad \frac{}{Time\ t \in DM_A(M)}\ \text{TIME}$$

$$\frac{}{Agent\ a \in DM_A(M)}\ \text{AGENT} \qquad \frac{}{Number\ n \in DM_A(M)}\ \text{NUMBER}$$

Fig. 3. Rules for $DM_A(M)$

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

$$\begin{aligned}\textbf{datatype}\ \mathit{event} = {} & \mathsf{Send}\ \mathit{transmitter}\ \mathit{msg}\ (\mathit{msg}\ \mathit{list}) \\ | \ & \mathsf{Recv}\ \mathit{receiver}\ \mathit{msg} \\ | \ & \mathsf{Claim}\ \mathit{agent}\ \mathit{msg}\end{aligned}$$

A *trace* is a list of timed events, where a timed event $(t, e) \in \mathbb{R} \times \mathit{event}$ pairs a timestamp with an event. Events are associated to agents and thereby to the agent's location. This association is either direct (*Claim*-events) or indirect via the association of transceivers with agents (*Send*- and *Recv*-events). The timed event $(t_S, \mathit{Send}\ \ \mathit{Tx}_A^i\ \ m\ L)$, for example, denotes that agent A has sent a message m using his transmitter Tx_A^i at time t_S and has associated the protocol data L with the event. The list of messages L models local state information and can contain messages used to construct m and times of preceding events. Such a *Send*-event may induce multiple *Recv*-events of the form $(t_R, \mathit{Recv}\ \mathit{Rx}_B^j\ m)$, where the timestamps t_R and the receivers Rx_B^j must be consistent with the network topology.

A *Claim*-event models a belief or a conclusion made by a protocol participant, formalized as a message. For example, after successfully completing a run of the authenticated ranging protocol (from Section 3.1) with Bob, Alice concludes at some time t_C that d_{AB} is an upper bound on her distance to Bob. We model this by adding the event $(t_C, \mathit{Claim}\ A\ (B, d_{AB}))$ 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.

Note that the timestamps used in both traces and rules use the notion of absolute time. However, agents' clocks may deviate arbitrarily from absolute time. We must therefore translate the absolute timestamps to model agent's local views. We describe this translation in Section 4.4.

Knowledge and Used Messages. Each agent A holds some initial knowledge, denoted $\mathit{initKnows}_A$, which depends on the executed protocol. In a system run with trace tr, the knowledge of an agent A is defined as the union of the initial knowledge and all received messages.

$$\mathit{knows}_A(tr) \equiv \{m \mid \exists\, k\ t.(t,\ \mathit{Recv}\ \ \mathit{Tx}_A^k\ \ m) \in tr\} \ \cup\ \mathit{initKnows}_A$$

From the known messages, A can derive all messages in $\mathit{DM}_A(\mathit{knows}_A(tr))$.

For a given term m, the subterm relation $\ll$ and the extractable subterm relation $\sqsubseteq$ are defined inductively by the rules in Figure 4. We use $\ll$ to define the set of messages appearing in a trace tr.

$$\mathit{used}(tr) \equiv \{n \mid \exists\, A\ k\ t\ m.(t, \mathit{Send}\ \mathit{Tx}_A^k\ m) \in tr\ \wedge\ n \ll m\}$$

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 $m \notin \mathit{used}([a_1, \ldots, a_{i-1}])$ and $m \in \mathit{used}([a_1, \ldots, a_i])$. In other words, a_i is the first event where m appears.

4.4 Network, Intruder, and Protocols

We now describe our rules that inductively define the set of traces $\mathit{Tr}(\mathit{proto})$ for a system parameterized by a protocol *proto*. The base case, modeled by the Nil rule

$$\frac{}{m \ll m} \qquad \frac{(a,b) \ll m}{a \ll m} \qquad \frac{(a,b) \ll m}{b \ll m} \qquad \frac{Hash\ \ c \ll m}{c \ll m} \qquad \frac{\{c\}_k \ll m}{c \ll m} \qquad \frac{\{c\}_k \ll m}{k \ll m}$$

$$\frac{}{m \sqsubseteq m} \qquad \frac{(a,b) \sqsubseteq m}{a \sqsubseteq m} \qquad \frac{(a,b) \sqsubseteq m}{b \sqsubseteq m} \qquad \frac{\{c\}_k \sqsubseteq m}{c \sqsubseteq m}$$

Fig. 4. Rules for $\ll$ and $\sqsubseteq$

$$\frac{}{[\,] \in Tr(proto)}\ \textsc{Nil}$$

$$\frac{\begin{array}{c} tr \in Tr(proto) \qquad t_R \geq maxtime(tr) \\ (t_S, Send\ \ Tx^i_A\ \ m\ L) \in tr \\ cdist_{Net}(Tx^i_A, Rx^j_B) = t_{AB} \\ t_{AB} \neq \bot \qquad t_R \geq t_S + t_{AB} \end{array}}{tr.(t_R, Recv\ \ Rx^j_B\ \ m) \in Tr(proto)}\ \textsc{Net} \qquad \frac{\begin{array}{c} tr \in Tr(proto) \qquad t \geq maxtime(tr) \\ m \in DM_{I_A}(knows_{I_A}(tr)) \end{array}}{tr.(t, Send\ \ Tx^k_{I_A}\ \ m\ [\,]) \in Tr(proto)}\ \textsc{Fake}$$

$$\frac{\begin{array}{c} tr \in Tr(proto) \qquad t \geq maxtime(tr) \qquad step \in proto \\ (act, m) \in step(view(H_A, tr), H_A, ctime(H_A, t)) \qquad m \in DM_{H_A}(knows_{H_A}(tr)) \end{array}}{tr.(t, translateEv(H_A, act, m)) \in Tr(proto)}\ \textsc{Proto}$$

Fig. 5. Rules for $Tr(proto)$

in Figure 5, states that the empty trace is a valid trace for all protocols. The other rules describe how valid traces can be extended. The rules model the network behavior, the possible actions of the intruders, and the actions taken by honest agents following the protocol.

Network Rule. The NET-rule models message transmission from transmitters to receivers, constrained by the network topology. 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 between these events is bounded by the communication distance between the transmitter and the receiver.

If there is a *Send*-event in the trace tr and the NET-rule's premises are fulfilled, a corresponding *Recv*-event is appended to the trace. The restriction on connectivity and 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 transmitter, t_S is the sending time, and t_R is the receiving time.

Note that one *Send*-event can result in multiple *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 the reflection of the signal carrying the message. Moreover, 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 an intruder, resulting in message loss, are modeled by traces where the NET-rule is not applied for a given *Send*-event and receiver, even if all premises are fulfilled.

The timestamps associated with *Send*-events and *Recv*-events denote the starting times of message transmission and reception. Thus, our network rule captures the

latency of links, but not the message-transmission time, which also depends on the message's size and the transmission speed of the transmitter and the receiver. Some implementation-specific attacks, such as those described in [Sastry et al. 2003; Clulow et al. 2006], are therefore not captured in our model.

The premise $t \geq maxtime(tr)$, included in every rule except NIL, ensures that timestamps increase monotonically within each trace. Here t denotes the timestamp associated with the new event and $maxtime(tr)$ denotes the latest timestamp in the trace tr. This premise guarantees that the partial order on events induced by their timestamps is consistent with the order of events in the list representing the trace. However, events can happen at the same time.

Intruder Rule. The FAKE-rule in Figure 5 describes the intruders' behavior. An intruder can send any message m derivable from his knowledge. Intruders internal state does not need to be modeled since they behave arbitrarily.

Since knowledge is distributed, we use explicit *Send*-events and *Recv*-events to model the exchange of information between colluding intruders. With an appropriate $cdist_{Net}$ 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 degrees 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.

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 consistent with the protocol specification.

There are two types of possible actions which model an agent either sending a message with a given transmitter id and storing the associated protocol data or making a claim.

$$\textbf{datatype}\ action = \mathsf{SendA}\ \mathbb{N}\,(msg\ list) \mid \mathsf{ClaimA}$$

Note that message reception is already modeled by the NET-rule.

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

$$translateEv(A, \mathsf{SendA}\ \ k\ L, m) = Send\ \ Tx^k_A\ \ m\ L$$
$$translateEv(A, \mathsf{ClaimA}\ , m) = Claim\ A\ m$$

A protocol *step* is therefore of type $agent \times trace \times \mathbb{R} \rightarrow (action \times msg)\ set$. 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., $occursAt(Send\ \ Tx^i_A\ \ m\ L) = A$.

$$view(A, tr) = [(ctime(A, t), ev)\ |(t, ev) \in tr \wedge occursAt(ev) = A]$$

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

Using the above definitions, in Figure 5 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 tr at a given time t. If all premises are met, the PROTO-rule appends the translated event to the trace. Note that agents' behavior, modeled by the function *step*, is based only on the local clocks of the agents, i.e., agents cannot access the global time. Moreover, the restriction that all messages must be in $DM_{H_A}(knows_{H_A}(tr))$ ensures that agents only send messages derivable from their knowledge.

4.5 Protocol-Independent Results

Since the set of traces $Tr(proto)$ is parameterized by the protocol description *proto*, our model allows us to establish protocol-independent results that hold for all or only certain types of protocol. In this section, we present four lemmas about the origin of messages that we will use later when we analyze concrete protocols. The proofs presented in this section follow the formal proofs of the corresponding lemmas as they can be found in our Isabelle/HOL formalization.

Our 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. Note that according to the NONCE rule in Figure 3, agents can only derive nonces tagged with their own identity and all other nonces must be received over the network.

LEMMA 4.1. *Let A be an arbitrary (honest or dishonest) agent and let $(t_S^A, Send\ \ Tx_A^i\ \ m_A\ \ L_A)$ be the first event in the trace tr with $N \ll m_A$ for the nonce N. If there is another event $(t_S^B, Send\ \ Tx_B^j\ \ m_B\ L_B) \in tr$ with $A \neq B$ such that $N \ll m_B$, then $t_S^B - t_S^A \geq cdist_{LoS}(A, B)$.*

PROOF. We prove this by induction on the set of valid traces. The proof for the rules NIL and NET follows trivially from the induction hypothesis since these rules do not introduce *Send*-events. We therefore consider the two remaining rules that add *Send*-events to traces. Let A be an agent and $(t_S^A, Send\ \ Tx_A^i\ \ m_A\ L)$ be the first event that contains the nonce N as a subterm.

FAKE**:** Assume that an event $(t_S^I, Send\ \ Tx_I^k\ \ m_I\ \ [])$ is appended to the trace. The only interesting cases are the ones where $A \neq I$ and where $N \ll m_I$. From the premises of the rule, we know $m_I \in DM_I(knows_I(tr))$. Since I cannot guess a nonce created by A, I must have received a message containing N at time t_R^I, where $t_R^I \leq t_S^I$. Since every *Recv*-event is preceded by a corresponding *Send*-event, there must be an event in the trace occurring at some agent C at time t_S^C, where $t_S^C \leq t_R^I - cdist_{Net}(Tx_C^l, Rx_I^h)$. From the induction hypothesis, we have $t_S^C - t_S^A \geq cdist_{LoS}(A, C)$. Using the triangle inequality for the physical distance and the consistency condition forbidding faster-than-light communication, $t_S^I - t_S^A \geq cdist_{LoS}(A, I)$ immediately follows.

PROTO**:** The event $(t_S^B, translateEv(B, action, m_B))$ is appended to the trace. Only the case where $action = SendA\ \ tid\ L$, $A \neq B$, and m_B contains N is interesting. From the premises 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. □

The next lemma is similar to Lemma 4.1 and concerns the earliest possible time when an agent can receive a nonce.

LEMMA 4.2. *Let A be an agent and let $(t_S^A, Send\ \ Tx_A^i\ \ m_A\ L_A)$ be the first event in the trace tr with $N \ll m_A$. If tr contains an event $(t_R^B, Recv\ \ Rx_B^j\ \ m_B)$ where $N \ll m_B$, then $t_R^B - t_S^A \geq cdist_{LoS}(A, B)$ holds.*

PROOF. We prove this by induction on the set of valid traces, considering the individual rules that introduce *Recv*-events; rules that introduce *Claim*-events or *Send*-events are therefore not of interest. Thus the only rule we need to consider is the NET-rule.
NET**:** We know that every *Recv*-event $(t_R^B, Recv\ \ Rx_B^k\ \ m)$ introduced by the *Net*-rule must be preceded by a corresponding *Send*-event $(t_S^C, Send\ \ Tx_C^l\ \ m\ L)$, where $t_S^C \leq t_R^B - cdist_{Net}(Tx_C^l, Rx_B^k)$. From Lemma 4.1, we know that $t_S^C - t_S^A \geq cdist_{LoS}(A, C)$ for the event that generates the nonce at time t_S^A. Reordering the first inequality, we get $cdist_{Net}(Tx_C^l, Rx_B^k) \leq t_R^B - t_S^C$. Using the consistency condition on physical and communication distance, we conclude $cdist_{LoS}(C, B) \leq t_R^B - t_S^C$. Using both inequalities, we get $t_R^B - t_S^A = t_R^B - t_S^C + t_S^C - t_S^A \geq cdist_{LoS}(A, C) + cdist_{LoS}(C, B)$. By the triangle equality for physical distances, $(cdist_{Net}(A, B) \leq cdist_{LoS}(A, B) + cdist_{Net}(B, C))$, we get the desired inequality $t_R^B - t_S^A \geq cdist_{LoS}(A, B)$, proving the claim of the theorem. □

The next lemma concerns with signatures and their creation time.

LEMMA 4.3. *Let A be an honest agent and let $(t_S^B, Send\ \ Tx_B^i\ \ m_B\ L) \in tr$ be an event in the trace tr where $\{m\}_{SK_A} \ll m_B$ for some message m. Then there is a send event $(t_S^A, Send\ \ Tx_A^j\ \ m_A\ L') \in tr$ such that $\{m\}_{SK_A} \ll m_A$ and $t_S^B - t_S^A \geq cdist_{LoS}(A, B)$.*

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

PROOF SKETCH. The proof is analogous to the proof of Lemma 4.1, but additionally uses the fact that intruders cannot create signatures on behalf of honest agents since the signing keys of honest agents are never leaked. □

A similar lemma (Lemma 4.4) also holds for *MAC*s.

LEMMA 4.4. *Let A and B be honest agents and C a different, possibly dishonest, agent. Furthermore let $(t_S^C, Send\ \ Tx_C^i\ \ m_C\ L)$ be an event in the trace tr where the message $MAC_{K_{AB}}(m) \ll m_C$ for some m and a shared secret key K_{AB}. Then for E either equal to A or B, there is a Send event $(t_S^E, Send\ \ Tx_E^j\ \ m_E\ L') \in tr$ where $MAC_{K_{AB}}(m) \ll m_E$ and $t_S^C - t_S^E \geq cdist_{LoS}(E, C)$.*

5. APPLYING THE MODEL

In this section, we use our model to analyze the security properties of four protocols: authenticated ranging, ultrasonic distance-bounding, TESLA broadcast authentication, and a secure time synchronization protocol. 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. In the fourth example, we restrict ourselves to constant clock errors again.

5.1 Authenticated Ranging

To define the set of possible traces for the authenticated ranging protocol introduced in Section 3.1, we formalize the set of protocol steps $proto\_ar = \{ar_1, ar_2, ar_3\}$. Each step function $ar_i(tr, A, t)$ yields the possible actions of the agent A executing the protocol step i with his view of the trace tr at the local time t. We have formalized each step in Isabelle/HOL using set comprehension, but present the steps here as rules for readability. For each rule r, the set we define by comprehension is equivalent to the set defined inductively by the rule r.

1) An honest agent A can start a protocol run by sending a fresh nonce N_A as a challenge. We use the index r to denote radio transmitters and receivers of honest agents. Note that dishonest agents have similar transceivers, but do not need the protocol rules to participate in protocol runs since they can create *Send*-events for any message that is derivable from their knowledge.

$$\frac{N_A \notin used(tr)}{(SendA\ \ r\ \ [\,], N_A) \in ar_1(tr, A, t_S^A)}$$

2) An honest agent receiving a challenge message may respond with the corresponding message.

$$\frac{(t_R^B, Recv\ \ Rx_B^r\ \ N_A) \in tr}{(SendA\ \ r\ \ [\,], \{N_A, t_S^B - t_R^B\}_{SK_B}) \in ar_2(tr, B, t_S^B)}$$

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

$$\frac{\begin{array}{c}(t_S^A, Send\ \ Tx_A^r\ \ N_A\ \ [\,]) \in tr \\ (t_R^A, Recv\ \ Rx_A^r\ \{N_A, \delta\}_{SK_B}) \in tr\end{array}}{(ClaimA, (A, (t_R^A - t_S^A - \delta) * \frac{c}{2})) \in ar_3(tr, A, t)}$$

The premises state that A has initiated a protocol run and received a response from agent B. A therefore believes that $(t_R^A - t_S^A - \delta) * \frac{c}{2}$ is an upper bound on the distance to B.

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 for this protocol. In the rest of the section, we consider the set of all traces $Tr(proto\_ar)$ corresponding to the authenticated ranging protocol.

Security Analysis. As explained in Section 3.1, 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 5.1. *Let A and B be honest agents, $tr \in Tr(proto\_ar)$, and $(t, Claim\ A\ (B,d)) \in tr$, then $d \geq |loc_A - loc_B|$.*

The theorem states that whenever the authenticated ranging protocol has been successfully executed between a pair of honest agents, then the resulting conclusion about their mutual distance is a upper bound on the physical distance between the agents.

For our proof, we use three of our protocol-independent lemmas about message ordering from Section 4 and the fact that δ sent in the second protocol message is always equal to $\delta = t_B^S - t_B^R$, the delay between the *Recv*-event and *Send*-event, provided B is honest. This follows directly from the definition of ar_2.

PROOF. Since only the step ar_3 adds events of the form $(t_C^A, Claim\ A\ \ (B,d))$, we know from the premises of ar_3 that N_A originates at the event $(t_S^A, Send\ \ Tx_A^r\ \ N_A\ [])$ in the trace. Furthermore, there is an event $(t_R^A, Recv\ \ Rx_A^r\ \ \{N_A, \delta\}_{SK_A})$ and $d = \frac{c}{2} * (t_R^A - t_S^A - \delta)$.

From the above, it follows that there is a *Send*-event in the trace tr at a time t_S^C, with $t_R^A - t_S^C \geq cdist_{LoS}(C, A)$, produced by some possibly dishonest agent. Using Lemma 4.3, we conclude that B sent a message containing the signature at time t_S^B, where $t_S^C - t_S^B \geq cdist_{LoS}(B, C)$. This message must result from an application of the rule PROTO with step ar_2, since B is assumed to be honest. Hence there must exist a *Recv*-event at B at time t_R^B and $\delta = t_S^B - t_R^B$. Finally we use Lemma 4.2 to show that $t_R^B - t_S^A \geq cdist_{LoS}(A, B)$ and sum up the inequalities.

$$\begin{aligned} t_R^A - t_S^A - \delta &= t_R^A - t_S^C + t_S^C - t_S^B + t_R^B - t_S^A \\ &\geq cdist_{LoS}(C, A) + cdist_{LoS}(B, C) + cdist_{LoS}(A, B) \\ &\geq cdist_{LoS}(B, A) + cdist_{LoS}(A, B) \\ &= 2 * cdist_{LoS}(A, B) \end{aligned}$$

Therefore we conclude $d = \frac{c}{2} * (t_R^A - t_S^A - \delta) \geq c * cdist_{LoS}(A, B) = |\, loc_A - loc_B\,|$. □

The specification and verification of this protocol was relatively straightforward since the protocol independent lemmas could be directly used to obtain the required inequalities on the times.

5.2 Ultrasound Distance Bounding

In our second example, we consider a protocol for *distance bounding* that uses radio signals as well as ultrasound signals to exchange messages between the communicating parties. The goal of the protocol presented in Figure 6 is for the verifier Alice to determine an upper bound on the distance to a possibly dishonest prover 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 $s * (t_R^A - t_S^A)$ on the distance, where s denotes the speed of sound. 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 required for signing the response. Furthermore, by using ultrasound, the protocol can be implemented on off-the-shelf devices because time measurements with nanosecond precision are not required. This type of protocol has been proposed

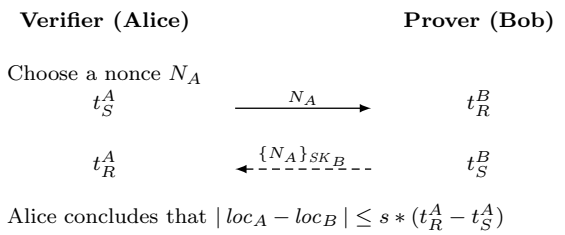


Fig. 6. Distance Bounding Protocol (dashed arrow denotes ultrasound transmissions)

in [Sastry et al. 2003] to enable Alice to verify a location claim of Bob (acting as the prover).

We assume that all agents A are equipped with ultrasound transmitters Tx_A^{us} and receivers Rx_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.

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

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

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

Note that it has recently been shown in [Rasmussen et al. 2009] that radio signals may induce a current in audio receiver circuits. More precisely, the authors of that paper demonstrated that this technique allows to successfully trigger receive events on ultrasound receivers using radio signals. This would enable trivial attacks against the protocol under consideration. The fact that we need this explicit additional assumption in our model to successfully prove the security of the protocol shows that our model accounts for subtle problems of this kind. However, we shall assume from now on that the countermeasures described in [Rasmussen et al. 2009] have been implemented and we can keep the assumption.

We now give the set of step functions $proto\_db = \{db_1, db_2, db_3\}$ defining the distance bounding protocol.

1) The start step db_1 initiates a protocol run.

$$\frac{N_A \notin used(tr)}{(SendA\ \ r\ \ [\,], N_A) \in db_1(A, tr, t_S^A)}$$

2) The reply step db_2 models the behavior of provers that respond to initial messages according to the protocol specification. Note that the ultrasound transmitter Tx_B^{us} is selected for the reply.

$$\frac{(t_R^B, Recv\ \ Rx_B^r\ \ N_A) \in tr}{(SendA\ \ us\ \ [\,], \{N_A\}_{SK_B}) \in db_2(B, tr, t_S^B)}$$

3) The final step db_3 introduces a *Claim*-event when a verifier A receives a response to his initial challenge on his ultrasound receiver Rx^{us}_A.

$$\frac{(t^A_S, Send\ \ Tx^r_A\ \ N_A\ \ [\,]) \in tr \quad (t^A_R, Recv\ \ Tx^{us}_A\ \ \{N_A\}_{SK_B}) \in tr}{(ClaimA, (B, (t^A_R - t^A_S) * s)) \in db_3(A, tr, t^A_C)}$$

This models what A concludes about a signal that apparently traveled from B to A using the speed of sound s in the time $t^A_R - t^A_S$. Namely A concludes that $s * (t^A_R - t^A_S)$ is a reliable upper bound on the distance to B.

Security Analysis. The security property to be achieved by the distance bounding protocol is similar to the property of the authenticated ranging protocol proved in Theorem 5.1. Since the prover's computation time is not used in computing the distance, the protocol does not require the prover to be honest. We would therefore expect a statement like the following to hold:

PROPOSITION 5.2. *Let A be an honest agent and B be any agent. Furthermore consider a valid trace $tr \in TR(proto\_db)$, where $(t, Claim\ A\ (B, d)) \in tr$. Then $d \geq |\, loc_A - loc_B\,|$, i.e., the distance measured by the protocol is an upper bound on the physical distance between the involved agents.*

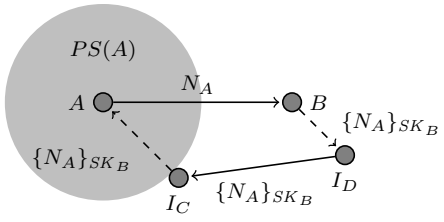


Fig. 7. Attack on DB using Ultrasound

However, as shown in [Sedihpour et al. 2005], this proposition is false without any further assumptions, as the attack in Figure 7 involving two colluding intruders shows. In the figure, $PS(A)$ denotes the private space of A and is defined as the largest circle centered at A such that A can ensure that no intruder is inside. To mount the attack, I_D is placed close to B and receives B's reply over ultrasound. I_D then uses a fast radio link to forwards it to the second intruder I_C who close to A. I_C finally delivers the message to A using ultrasound. We have proven in our Isabelle/HOL formalization that this attack is captured in our model by showing that the corresponding attack trace is a valid trace. The inequality involving the communication distances necessary for such an attack to work is

$$\begin{aligned} |\, loc_A - loc_B\,|/s > {} & cdist_{Net}(Tx^r_A, Rx^r_B) + cdist_{Net}(Tx^{us}_B, Rx^{us}_{I_D}) \\ & + cdist_{Net}(Tx^r_{I_D}, Rx^r_{I_C}) + cdist_{Net}(Tx^{us}_{I_C}, Rx^{us}_A)\,. \end{aligned}$$

If the inequality holds, intruders connected by a fast radio link can speed up ultrasound communication between A and B (using their radio link), such that the deduced distance is smaller than the real distance between A and B. This attack has been discovered and implemented in [Sedihpour et al. 2005].

In light of the above, we prove Proposition 5.2 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., [Sastry et al. 2003; Capkun and Cagalj 2006] for location-based access control and device pairing. It holds, for example, in environments where A can visually verify the absence of nearby intruders. The following inequality ensures that no intruder is closer to the verifier A than the possibly dishonest prover B: $\forall\, D.\, |\, loc_A - loc_{I_D} |\ \geq\ |\, loc_A - loc_B\, |$.

Note that this assumption thwarts Mafia frauds as well as Terrorist frauds (as defined in [Brands and Chaum 1994]). A mafia fraud is an attack where an intruder plays man-in-the-middle between a verifier and an honest prover. Since there is no intruder closer to the prover A than the verifier B, this kind of attack is impossible in our setting. Similarly, a terrorist fraud is an attack where an attacker plays man-in-the-middle between a verifier and a dishonest prover. This would require a second attacker being located closer to the verifier than the dishonest prover B; this setup also trivially violates the private space assumption of the verifier A.

We now restate Proposition 5.2, adding this additional assumption, and prove the result.

THEOREM 5.3. *Let A be an honest agent and B be any agent such that $\forall\, D.\, |\, loc_A - loc_{I_D} |\ \geq\ |\, loc_A - loc_B\, |$. Furthermore consider a valid trace $tr \in TR(proto\_db)$ where $(t_A^C, Claim\ A\ (B, d)) \in tr$. Then $d \geq |\, loc_A - loc_B\, |$.*

PROOF. We proved this by induction over traces, using Lemma 4.1. For the empty trace the claim is trivially true, so the base case for the induction holds. Since only the PROTO rule with step db_3 creates events of the form $(t_A^C, Claim\ A\ \ (B, d))$, we do not need to consider other rules. From the premises of db_3, we conclude that the nonce N_A originates at an event $(t_S^A, Send\ \ Tx_A^r\ \ N_A\ [\,])$. Furthermore, there must be an event $(t_R^A, Recv\ \ Rx_A^{us}\ \{N_A\}_{SK_B})$, such that $d = s * (t_R^A - t_S^A)$. Therefore we must show that $t_R^A - t_S^A \geq |\, loc_A - loc_B\, |/s$.

Since *Recv*-events are only introduced by applications of the NET-rule, we know that 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_S^B, with $t_R^A - t_S^B \geq cdist_{Net}(Tx_B^{us}, Rx_A^{us})$. From Lemma 4.1 it follows that $t_S^B \geq t_S^A$, since N_A is included in the message. Together with the previous inequality and using the assumption that messages received by ultrasound receivers do not travel faster than s, we conclude that $t_R^A - t_S^A \geq cdist_{Net}(Tx_B^{us}, Rx_A^{us}) \geq |\, loc_A - loc_B\, |/s$.

In the second case, the message is sent by the intruder I at time t_S^I. From the assumption that B is located in the private space of A, the distance between A and I is bounded below by the distance between A and B. Additionally, the assumptions ensure that a message received by Rx_A^{us} has not traveled with a speed faster than s. Together with $t_S^I \geq t_S^A$ (which follows from Lemma 4.1) this completes the proof as $t_R^A - t_S^A \geq t_R^A - t_S^I \geq cdist_{Net}(Tx_I^j, Rx_A^{us}) \geq |\, loc_A - loc_I\, |/s \geq |\, loc_A - loc_B\, |/s$. □

Note that our 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.

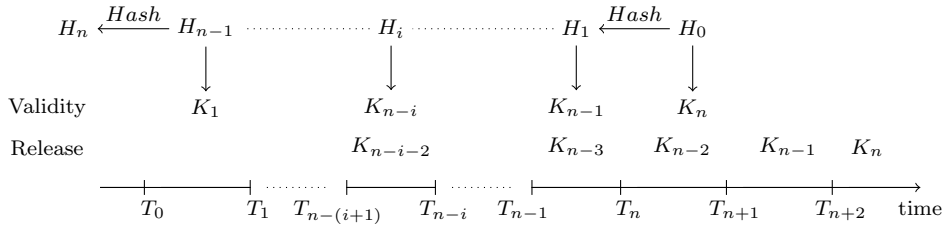


Fig. 8. Association of Hash Chain Elements to Validity ad Release Windows

5.3 A Delayed Key Disclosure Protocol

In our next example, we model and verify security properties of a *Delayed Key Disclosure* protocol. Such protocols are used for broadcast authentication in resource-constrained environments such as sensor networks, where asymmetric cryptography might not be 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 thereby proves the origin of the message.

We formalize the TESLA broadcast authentication protocol [Perrig and Tygar 2002] which is an example of such a protocol. In TESLA, the sender commits to a sequence 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 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 according to the release schedule. We use a key schedule that defines $[T_{i+1}, T_{i+2}[$ as K_i's release window. Figure 8 depicts the key schedule.

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 contained only in the broadcaster Br's initial knowledge and the initial knowledge of the other agents just contains H_n.

The set of steps functions $proto\_dkd = \{dkd_1, dkd_2\}$ formalizes the delayed key disclosure protocol.

1) The dkd_1 step function 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 $[T_{i-3}, T_{i-2}[$. For $i = 1$ and $i = 2$, the keys $K_{-1} = H_{n+1}$ and $K_0 = H_n$,

which can be already derived from the public knowledge, are released.

$$\frac{t_S^{Br} \in [T_{i-1}, T_i[\qquad i \geq 1}{(SendA\ r\ [], (MAC_{K_i}(m), K_{i-2})) \in dkd_1(Br, tr, t_S^{Br})}$$

2) The dkd_2 step function models the conclusion of an agent R who received a message m authenticated with the key K_i before its expiration at T_i, under the release schedule. In addition, the agent has received K_i at a later time point.

$$\frac{\begin{array}{c} t_1^R < T_i \quad i \leq n \\ (t_1^R, Recv\ \ Rx_R\ (MAC_{K_i}(m), K_{i-2})) \in tr \\ (t_2^R, Recv\ \ Rx_R\ (MAC_{K_{i+2}}(m'), K_i)) \in tr \end{array}}{(ClaimA, (m, i)) \in dkd_2(R, tr, t)}$$

Note that in the premises of dkd_2 we do not restrict the arrival time t_2^R of the released key K_i. 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. However, the premises require that the corresponding message authentication code $MAC_{K_i}(m)$ has been received in the validity window of the corresponding key, i.e., before T_i.

Security Analysis. A broadcast protocol achieves *T-authentication* [Schaller et al. 2007] if the protocol guarantees both message-origin authentication and 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 5.4. *Let tr be a valid trace. If $(t_C^R, Claim\ H_R\ \ (m, i)) \in tr$, then there is a $(t_S^{BR}, Send\ \ Tx_{Br}^r\ (MAC_{K_i}(m),\ K_{i-2})\ [\,]) \in tr$, where $t_S^{BR} \in [T_{i-1}, T_i[$.*

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 5.4 using two lemmas about the temporal secrecy of hash-chain elements.

The first lemma states that, prior to the release of a key by the broadcast source, no other agent can use the key.

LEMMA 5.5. *Suppose that $0 \leq l \leq n$, A is an agent other than Br, and tr is a valid trace. If $(t, Send\ \ Tx_A^i\ \ X\ L) \in tr$, where $K_l \sqsubseteq X$, or 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 as an extractable subterm, then $maxtime(tr) \geq T_{l+1}$.*

PROOF. We prove this by induction on traces. Since the NIL case is obvious, we now consider the three remaining rules.
FAKE**:** The event $(t_S^I, Send\ \ Tx_I^k\ \ X\ )$ is added to the trace tr. We must only consider the case $K_l \sqsubseteq X$ where $K_l \sqsubseteq DM_A(knows_A(tr))$ follows from the premises of the rule and therefore $maxtime(tr) \geq T_{l+1}$ from the induction hypothesis.
CON**:** The event $(t_R^A, Recv\ \ Rx_A^k\ \ X)$ is added to the trace tr. We must only consider the case where a message containing K_l becomes derivable by A. Hence $K_j \sqsubseteq X$ for some $j \geq l$ and there is a *Send*-event for X in tr as required by the premises of CON. The induction hypothesis can now be applied.

PROTO: The second step dkd_2 only adds a *Claim*-event, so we can concentrate on dkd_1. Here, the event $(t, Send\ \ Tx^r_{Br}\ (MAC_{K_i}(m), K_{i-2})\ [\,])$ is added to the trace tr. Note that $K_{i-2} \sqsubseteq (MAC_{K_i}(m), K_{i-2})$, but K_i is not an extractable subterm 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 5.6. *Suppose that tr is a valid trace and that* $(t^A_S, Send\ \ Tx^i_A\ \ M\ ) \in tr$, *where* $Hash(K_l, m) \sqsubseteq M$. *Furthermore suppose that* $maxtime(tr) < T_{l+1}$, *and* $0 < l < n$. *Then there exists an event* $(t^{Br}_S, Send\ \ Tx^r_{Br}\ (MAC_{K_l}(m), K_{l-2})\ ) \in tr$, *with* $t^{Br}_S \in [T_{l-1}, T_l[$.

PROOF. We must just consider the FAKE rule and the case where the event $(t^I_S, Send\ \ Tx^k_I\ \ X\ )$, 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 5.5, $maxtime(tr) \geq T_{l+1}$, 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 5.4 using the previous lemma is straightforward.

PROOF. Since only dkd_2 adds events of the form $(t, Claim\ H_R\ (m, i))$, we need not consider the other rules. From the premises of dkd_2, we conclude that there is a *Recv*-event with message $(MAC_{K_i}(m), K_{i-2})$ and time t^R_1, where $t^R_1 \in [T_{i-1}, T_i[$. 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 5.6, which completes the proof. □

After finding the right intermediate lemmas stating which messages must be secret during which time intervals, the proofs presented here are relatively straightforward and are mostly concerned with the cryptographic reasoning about the hash chain elements. In our Isabelle/HOL formalization, there is some additional complexity because we do not assume synchronized clocks.

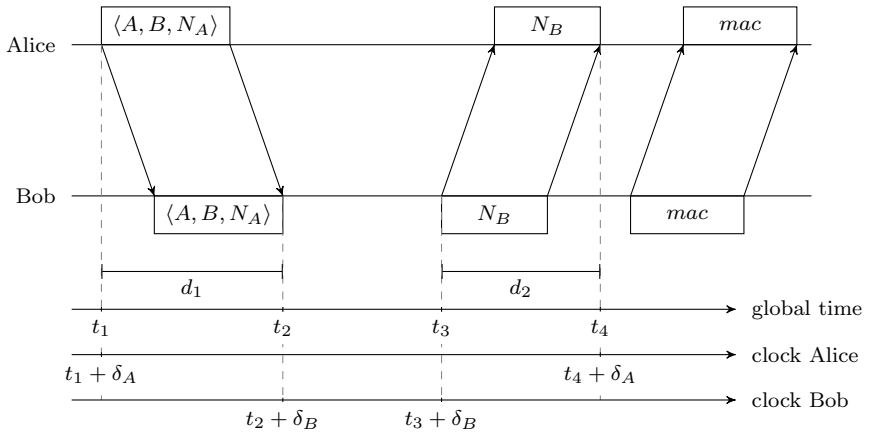
5.4 A Secure Time Synchronization Protocol

As a final example, we formalize a secure time synchronization protocol presented in [Ganeriwal et al. 2008]. The Enhanced Secure Pairwise Synchronization (E-SPS) protocol achieves clock synchronization between two honest nodes in the presence of external attackers by computing the relative clock offset between the two nodes.

We analyze this protocol under the following assumptions.

Constant clock offset. We assume constant clock offsets for each agent's clock during the execution of the protocol. Namely, for each agent A there is a δ_A such that $ctime(A, t) = t + \delta_A$.

Lower bound on message transmission time. We assume a maximal bandwidth for the connecting network. As a consequence, there is a lower bound d_{min} on the



If $d \leq d_{max}$, Alice accepts the last message and concludes that δ_{AB} is an approximation of the relative clock offset such that $|\delta_{AB} - (\delta_B - \delta_A)| \leq 2 * (d_{max} - d_{min})$, where

$$t_i^C = t_i + \delta_C$$
$$mac = MAC_{K_{AB}}(B, A, N_A, N_B, t_2^B, t_3^B)$$
$$d = ((t_2^B - t_1^A) + (t_4^A - t_3^B))/2$$
$$\delta_{AB} = ((t_2^B - t_1^A) - (t_4^A - t_3^B))/2$$

Fig. 9. Enhanced Secure Time Synchronization (E-SPS) Protocol

message transmission time for any message that contains a nonce. Even in the case of a malicious sender, it is impossible for honest agents to complete reception of a nonce before $t_{start} + d_{min}$, where t_{start} is the start time of the reception.

Upper bound on end-to-end delay. Finally, we assume that there is a maximal end-to-end delay when receiving a message containing a nonce. The maximal delay d_{max} consists of (1) the time for media access, (2) the time-of-flight, and (3) the message transmission time. In terms of media access, we assume that the hardware is capable of obtaining and inserting timestamps of reception and send events. Therefore we do not have to account for (1) since it does not affect the measurements. We obtain a bound for (2) by assuming a maximal distance between nodes. However, in most cases (2) is negligible compared to (3). For (3), it is possible to define a minimal transmission rate and to use the corresponding maximal delay. A detailed breakdown of the times can be found in [Ganeriwal et al. 2008].

Figure 9 depicts the E-SPS protocol as a sequence diagram, where time passes from left to right. The protocol securely computes an approximation δ_{AB} of the relative clock offset $\delta_B - \delta_A$. To prove that the approximation error is bounded by $2*(d_{max} - d_{min})$, we formalize the protocol as the set of step functions $proto\_esps = \{esps_1, esps_2, esps_3, esps_4\}$. Recall that step functions involve only local times of agents, i.e., timestamps associated with trace events are translated to the local time of the corresponding agent.

1) The first step $esps_1$ models an initiator A sending an initial challenge. Here t_1^A

denotes the local time measured by A, corresponding to the global time $t_1^A - \delta_A$.

$$\frac{N_A \notin used(tr)}{(SendA\ r\ [\,], \langle A, B, N_A\rangle) \in esps_1(A, tr, t_1^A)}$$

2) The second step $esps_2$ models an agent B responding to an initial challenge.

$$\frac{\begin{array}{c} N_B \notin used(tr) \\ t_3^B \geq t_2^B \quad t_2^B \geq t_{2,start}^B + d_{min} \\ (t_{2,start}^B, Recv\ B\ \langle A, B, N_A\rangle) \in tr \end{array}}{(SendA\ r\ [A, N_A, t_2^B], N_B) \in esps_2(B, tr, t_3^B)}$$

Reception of the challenge starts at $t_{2,start}^B$ and ends d_{min} time units later at the earliest. B associates the protocol data A, N_A, and t_2^B with the resulting *Send*-event. Recall that all times refer to B's local clock.

3) The third step $esps_3$ models B sending the MAC.

$$\frac{(t_3^B, Send\ B\ N_B\ \ [A, N_A, t_2^B]) \in tr}{(SendA\ r\ [\,], MAC_{K_{AB}}(B, A, N_A, N_B, t_2^B, t_3^B)) \in esps_3(B, tr, t^B)}$$

Besides the exchanged nonces, B includes the times t_2^B and t_3^B, denoting the time when reception of the initial challenge ended and the time when the first challenge has been sent.

4) Finally, step $esps_4$ models A's reception of the responses.

$$\frac{\begin{array}{c} (t_1^A, Send\ A\ \langle A, B, N_A\rangle\ ) \in tr \\ (t_{4,start}^A, Recv\ A\ N_B) \in tr \\ (t_5^A, Recv\ A\ MAC_{K_{AB}}(B, A, N_A, N_B, t_2^B, t_3^B)) \in tr \\ d = ((t_2^B - t_1^A) + (t_4^A - t_3^B)/2 \\ \delta_{AB} = ((t_2^B - t_1^A) - (t_4^A - t_3^B))/2 \\ t_4^A \geq t_{4,start}^A + d_{min} \quad d \leq d_{max} \quad t \geq t_4^A \end{array}}{(ClaimA, (B, \delta_{AB})) \in esps_4(A, tr, t^A)}$$

If A concludes a clock-offset δ_{AB}, A must have received the first response and the corresponding MAC according to the protocol specification. A then computes the delay d and the offset δ_{AB} using his own time measurements and the timestamps received in messages from B (see Figure 9). Finally, A completes the protocol only if $d \leq d_{max}$. Otherwise, the end-to-end delay for the relevant transmissions took too long, which would result in an unreliable estimation of the clock offset.

First note that if neither the challenge nor the rapid-response is delayed by an intruder by or the environment, then $d_1 \approx d_2$ for the two transmission delays (see Figure 9) and $d = ((d_1 + \delta_B - \delta_A) + (d_2 + \delta_A - \delta_B))/2 = (d_1 + d_2)/2 \leq d_{max}$. In this case, Alice computes the relative clock offset $\delta_{AB} = ((d_1 - d_2) + 2 * \delta_B - 2 * \delta_A)/2 \approx \delta_B - \delta_A$. The upper bound d_{max} bounds the error that the adversary can introduce by delaying either the challenge or the rapid-response.

We first show that if there is a *Claim* event, then there are four corresponding times that satisfy the following (in)equalities.

LEMMA 5.7. *Let A and B be honest agents, $tr \in Tr(proto\_esps)$, and $(t, Claim\ A\ (B, \delta_{AB})) \in tr$. Then there are times t_1, t_2, t_3, and t_4 such that the following hold:*

(1) $d_{min} \leq t_2 - t_1$
(2) $t_2 \leq t_3$
(3) $d_{min} \leq t_4 - t_3$
(4) $\delta_{AB} = (((t_2 + \delta_B) - (t_1 + \delta_A)) - ((t_4 + \delta_A) - (t_3 + \delta_B)))/2$
(5) $((t_2 - t_1) - (t_4 - t_3))/2 \leq d_{max}$

PROOF. From the existence of the *Claim* event for the honest agent A in the trace, we can conclude that all the premises of $esps_4$ hold. There is a *Send* event of a fresh nonce N_A at a time t_1^A. There is a *Recv* event of a nonce N_B that starts at a time $t_{4,start}^A$ and is completed at a time t_4^A, where $t_4^A - t_{4,start}^A \geq d_{min}$. Finally, there is a *Recv* event of $MAC_{K_{AB}}(B, A, N_A, N_B, t_2^B, t_3^B)$. For the global times $t_1 = t_1^A - \delta_A$, $t_4 = t_4^A - \delta_A$, $t_{4,start} = t_{4,start}^A - \delta_A$, $t_2 = t_2^B - \delta_B$, and $t_3 = t_3^B - \delta_B$ we can conclude that (4) and (5) hold and $t_4 - t_{4,start} \geq d_{min}$.

The MAC must originate from B by Lemma 4.4 because the key is assumed to be a shared secret between A and B and $esps_3$ is the only protocol step that uses the key. Then $esps_2$ must have also been executed by B and there must be a *Send* event of the nonce N_B at the global time t_3 defined above and the reception of the nonce N_A must have started at a global time $t_{2,start}$ and completed at the global time t_2 defined above such that $t_2 - t_{2,start} \geq d_{min}$ and (2) hold. From Lemma 4.2, applied to the reception and sending of the nonces N_A and N_B, we obtain $t_1 \leq t_{2,start}$ and $t_3 \leq t_{4,start}$. Together with $t_2 - t_{2,start} \geq d_{min}$ and $t_4 - t_{4,start} \geq d_{min}$, we obtain (1) and (3), which completes the proof. □

Using these inequalities, it is easy to prove the following security property already stated in Figure 9.

THEOREM 5.8. *Let A and B be honest agents, $tr \in Tr(proto\_esps)$, and $(t, Claim\ A\ (B, \delta_{AB})) \in tr$, then $|\delta_{AB} - (\delta_B - \delta_A)| \leq 2 * (d_{max} - d_{min})$.*

PROOF. Using Lemma 5.7, we obtain t_1, t_2, t_3, and t_4 such that the corresponding (in)equalities hold. Then $\delta_{AB} - (\delta_B - \delta_A) = ((t_2 - t_1) - (t_4 - t_3))/2$ and both $(t_2 - t_1)$ and $(t_4 - t_3)$ are greater or equal to d_{min}. If $(t_2 - t_1) \leq (t_4 - t_3)$, then $(t_4 - t_3) \leq 2 * d_{max} - d_{min}$ and therefore $(t_4 - t_3) - (t_2 - t_1) \leq 2 * (d_{max} - 2 * d_{min})$, as desired. The other case is analogous. □

Aside from the reasoning about the creation of MACs, which is covered by our protocol independent lemmas, most of the theorem-proving work concerns establishing the desired inequalities between the times of the different events. Most of these inequalities stem from the relation between the different clock offsets of the two nodes.

5.5 Summary

The four case studies demonstrate that our model captures sufficient details of the physical environment to enable the formalization of a wide range of protocols

for wireless networks. It also allows us to formulate the security properties and additional assumptions on the environment in a natural way.

Our security proofs show that our model is nevertheless abstract enough to allow fully mechanized proofs with reasonable time and effort. The authenticated ranging and the ultrasound distance bounding case studies were developed in parallel with our framework and hence we can only give the combined time which was in the range of months. The verification of TESLA and E-SPS used the infrastructure and took two weeks, respectively three days. We note that verifying physical protocols is not substantially harder than verifying classical protocols. This is because reasoning about message derivation is mostly orthogonal to reasoning about time and location, we can therefore use standard techniques.

In the process of mechanizing the security proofs, we discovered some necessary assumptions about the environment that we did not consider from the start. For example, during development we discovered that the additional requirements on ultrasound reception given in Section 5.2 were necessary to complete the verification of the distance bounding protocol.

6. ISABELLE/HOL FORMALIZATION

We briefly survey our Isabelle/HOL formalization [Schmidt and Schaller 2010]. Our model builds on the following theories, depicted in Figure 10 along with their dependencies.

Message Theory: Our message theory (Section 4.3) models a free term algebra and is based on Paulson's work [Paulson 1998]. It also includes a formalization of hash chains and their properties. We have extended this message theory to handle the XOR operator and its algebraic properties and applied the extended model to the analysis of a class of distance bounding protocol in [Basin et al. 2009].

Geometric Properties of $\mathbb{R}^3$*:* Since agents' locations are vectors in $\mathbb{R}^3$ (Section 4.2), we use the formalization of real numbers provided in Isabelle's standard library.

Parameterized Communication Systems: Rules (Section 4.4) describe the network properties, possible intruder actions, and the protocol steps. Together these inductively define the set of possible traces.

Protocol Formalizations: These are given by sets of step functions (Section 5), formalizing the actions taken by agents running the protocol. For a given protocol, we instantiate the 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.

Protocol Independent Properties: Parameterizing the set of possible traces by protocol step functions allows us to prove protocol independent system properties as described in Section 4.5.

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 resembling the proofs presented in this paper. Figure 11 consists of a table that compares the sizes of the different parts of the formalization.

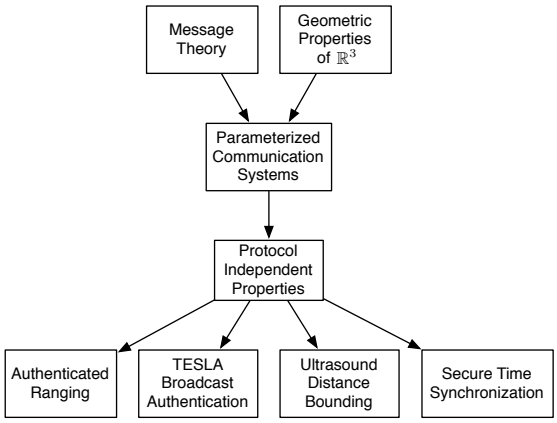


Fig. 10. Dependency Graph of our Isabelle Theory Files

Theory	Lines of Code	Lemmas	Definitions	Pages
Message Theory	2211	229	18	43
Geometric Properties of $\mathbb{R}^3$	272	13	6	6
Parameterized Communication Systems	633	51	4	12
Protocol Independent Properties	2342	57	4	39
Authenticated Ranging	489	8	6	11
Ultrasound Distance Bounding	681	15	6	15
TESLA Broadcast Authentication	883	9	8	18
Secure Time Synchronization	961	20	10	22
Total	8472	402	62	166

Fig. 11. Statistics about our Isabelle Formalization

7. 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 [Armando et al. 2005], and interactive methods, such as theorem proving [Paulson 1998]. In both settings, it is standard to formalize an intruder 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. Whereas the related works are restricted to specific types of protocols and address at most one or two of these aspects, our model combines all three aspects and is therefore applicable to a wider range of protocols. For example, there has been, to the best of our knowledge, no formal analysis of an ultrasound distance bounding protocol before. Such an analysis obviously requires a model that takes into account time, nodes' locations, and a network topology that reflects properties of different communication media.

Most approaches formalizing time [Delzanno and Ganty 2004; Evans and Schneider 2000] only focus on timestamps, which are used to reason about key-expiration, e.g., in protocols like Kerberos. Corin et al. [Corin et al. 2007] use timed automata [Alur and Dill 1994] to model timing attacks and timing issues like timeouts and retransmissions in security protocols. In [Gorrieri et al. 2003] 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 the messages sent by the broadcast source, abstracting away from the network and the topology. Archer uses the TAME [Archer 2000] interface to PVS in [Archer 2002] to prove the authenticity of messages received in the correct validity window of the corresponding key in TESLA. In [Hopcroft and Lowe 2004] the authors model two TESLA variants in CSP. Their formalization leads to a finite state space allowing for automatic verification using the model checker FDR.

Network topology has been considered in formal approaches for analyzing routing protocols in ad hoc networks [Acs et al. 2006; Nanz and Hankin 2006; Yang and Baras 2003; Arnaud et al. 2010]. Closely related is the notion of secure neighbor discovery (see for example [Papadimitratos et al. 2008]). In this setting, a node must detect its direct communication partners, for example, as a basis for topology information used for routing. It has been shown in [Poturalski et al. 2008] that under certain assumptions, there is no protocol that can achieve this objective.

Node location has been, to our knowledge, only used in informal proofs. For example, Sastry et al. [Sastry et al. 2003] propose a protocol for verifying location claims based on ultrasonic communication and provide an informal proof of its security and reliability. Avoine et al. [Avoine et al. 2010] present a framework for classifying different attack scenarios for distance bounding protocols. Other approaches only formalize the related notion of relative distance. In Meadows et al. [Meadows et al. 2006], 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.

In recent work [Basin et al. 2009], we have extended the message theory of the framework presented in this article to support the exclusive-or operator. We have used this extension to analyze a class of distance bounding protocols proposed in [Meadows et al. 2006].

8. CONCLUSION

We have presented a formal approach to modeling and verifying security protocols involving physical properties. Our model captures dense time, agent locations, and physical properties of the communication network. To our knowledge, this is the first formal model that combines 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 four different protocols and showed that our model captures relay attacks by distributed intruders.

REFERENCES

Acs, G., Buttyan, L., and Vajda, I. 2006. Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing 5,* 11, 1533–1546.

Alur, R. and Dill, D. 1994. A Theory of Timed Automata. *Theoretical Computer Science 126,* 2, 183–235.

Archer, M. 2000. Tame: Using pvs strategies for special-purpose theorem proving. *Ann. Math. Artif. Intell. 29,* 1-4, 139–181.

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

Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, J., Drielsma, P. H., Heám, P.-C., Kouchnarenko, O., Mantovani, J., Mödersheim, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Viganò, L., and Vigneron, L. 2005. The AVISPA tool for the automated validation of internet security protocols and applications. In *Proceedings of CAV'2005*. LNCS 3576. Springer-Verlag, 281–285.

Arnaud, M., Cortier, V., and Delaune, S. 2010. Modeling and verifying ad hoc routing protocols. *Computer Security Foundations Symposium, IEEE 0*, 59–74.

Avoine, G., Bingöl, M. A., Kardaş, S., Lauradoux, C., and Martin, B. 2010. A framework for analyzing RFID distance bounding protocols. *Journal of Computer Security – Special Issue on RFID System Security*.

Basin, D., Capkun, S., Schaller, P., and Schmidt, B. 2009. Let's get physical: Models and methods for real-world security protocols. In *TPHOLs '09: Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics*. Springer-Verlag, Berlin, Heidelberg, 1–22.

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

Capkun, S., Buttyan, L., and Hubaux, J.-P. 2003. 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*. ACM Press, New York, NY, USA, 21–32.

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

Capkun, S. and Hubaux, J. 2006. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications 24,* 2, 221–232.

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

Corin, R., Etalle, S., Hartel, P., and Mader, A. 2007. Timed analysis of security protocols. *Journal of Computer Security 15,* 6, 619–645.

Delzanno, G. and Ganty, P. 2004. Automatic verification of time sensitive cryptographic protocols. *Tools and Algorithms for the Construction and Analysis of Systems*, 342–356.

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

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

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

Francillion, A., Danev, B., and Capkun, S. 2010. Relay attacks on passive keyless entry and start systems in modern cars. In *Cryptology ePrint Archive: Report 2010/332*.

Ganeriwal, S., Pöpper, C., Capkun, S., and Srivastava, M. 2008. Secure time synchronization in sensor networks. *ACM Transactions on Information and System Security 11,* 4, 23.

Gorrieri, R., Martinelli, F., Petrocchi, M., and Vaccarelli, A. 2003. 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*. 139–154.

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

Hopcroft, P. J. and Lowe, G. 2004. Analysing a stream authentication protocol using model checking. *International Journal of Information Security 3,* 1, 2–13.

Kuhn, M. 2004. An asymmetric security mechanism for navigation signals. *IH 2004: 6th International Workshop on Information Hiding, Revised Selected Papers*, 239–252.

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

Meadows, C., Poovendran, R., Pavlovic, D., Chang, L., and Syverson, P. 2006. Distance bounding protocols: Authentication logic analysis and collusion attacks. *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, 279–298.

Nanz, S. and Hankin, C. 2006. A framework for security analysis of mobile wireless networks. *Theoretical Computer Science 367,* 1, 203–227.

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

Papadimitratos, P., Poturalski, M., Schaller, P., Lafourcade, P., Basin, D., Capkun, S., and Hubaux, J. 2008. Secure neighborhood discovery: a fundamental element for mobile ad hoc networking. *Communications Magazine, IEEE 46,* 2, 132–139.

Paulson, L. C. 1998. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security 6*, 85–128.

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

Poturalski, M., Papadimitratos, P., and Hubaux, J.-P. 2008. Secure neighbor discovery in wireless networks: formal investigation of possibility. In *ASIACCS '08: Proceedings of the 3rd ACM Symposium on Information, Computer, and Communications Security*. ACM, 189–200.

Rasmussen, K. B. and Capkun, S. 2010. Realization of RF distance bounding. In *Proceedings of the USENIX Security Symposium*.

Rasmussen, K. B., Castelluccia, C., Heydt-Benjamin, T. S., and Capkun, S. 2009. Proximity-based access control for implantable medical devices. In *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*. ACM.

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

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

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

Schmidt, B. and Schaller, P. 2010. Isabelle Theory Files: Formal Reasoning about Physical Properties of Security Protocols. http://people.inf.ethz.ch/benschmi/ProtoVeriPhy/.

Sedihpour, S., Capkun, S., Ganeriwal, S., and Srivastava, M. 2005. Implementation of attacks on ultrasonic ranging systems (demo). *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys)*.

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

Tippenhauer, N. O. and Capkun, S. 2009. Id-based secure distance bounding and localization. In *In Proceedings of ESORICS (European Symposium on Research in Computer Security)*.

Yang, S. and Baras, J. S. 2003. Modeling vulnerabilities of ad hoc routing protocols. In *SASN '03: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*. ACM, New York, NY, USA, 12–20.