Modeling and Verifying Physical Properties of Security Protocols for Wireless Networks

Modeling and Verifying Physical Properties of Security Protocols for Wireless Networks

Patrick Schaller, Benedikt Schmidt, David Basin, Srdjan Čapkun
Department of Computer Science ETH Zürich
Haldeneggsteig 4
8092 Zürich, Switzerland
{patrick.schaller, benedikt.schmidt, david.basin, srdjan.capkun} @inf.ethz.ch

ABSTRACT

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

1. INTRODUCTION

The shrinking size of microprocessors combined with the ubiquity of wireless network connections has led to new application areas for networked systems with novel security requirements for the protocols employed. Whereas "traditional" security protocols are mainly concerned with message secrecy or different 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 their physical distance. Such protocols may use authentication patterns combined with assumptions about the underlying communication medium, e.g., [9, 10, 26, 22].

Secure Localization: A node must determine its true location in an adversarial setting or make verifiable statements about its location by executing protocols with other nodes, e.g., [36, 23, 24, 34]. Both secure localization and physical proximity verification protocols, and attacks on them, have been implemented on RFID, smartcards and Ultra-Wide Band (UWB) platforms [18, 27, 35, 40].

Secure Time Synchronization: A node must securely synchronize its clock to the clock of another (trusted) node in an adversarial setting, e.g., [39, 20, 25]. These protocols also serve as a basis for more efficient secure networking protocols, e.g., for efficient broadcast authentication [31].

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

In this work, we present a formal model for reasoning about the security guarantees of protocols like those 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, corresponds to a set of nodes. The ability of the nodes to communicate and the speed of communication are determined by nodes' locations and by the propagation delays of the communication technologies they use. As a consequence, nodes (both honest and those controlled by the intruder) require time to share their knowledge and information cannot travel between nodes at speeds faster than the speed of light. The intruder and honest agents are, therefore, subject to physical restrictions. This results in a distributed intruder with restricted, but more realistic, communication capabilities than the classical Dolev-Yao intruder.

The main contribution of our work is the combination of a message and a communication model. Whereas the message model allows us to capture cryptographic aspects of protocol messages (under the assumption of perfect cryptography),

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

our communication model allows us to model relevant properties of the communication technology. Similar to Paulson's *Inductive Approach* [30], we have used Isabelle/HOL [29] to formalize our model and to prove security properties of the protocols presented in this paper. Our model reuses various parts of Paulson's formalization of the Dolev-Yao intruder in [30]. We model communication as send and receive events, where the communication technology and the network topology determine the parameters (e.g., time and location) of the receive event resulting from a given send event. As a proof of concept, we have formalized and verified three protocols. Their diverse features and properties reflect the broad scope of our model in applications where environmental factors and their physical constraints are used alongside cryptography to achieve security objectives.

The rest of the paper is organized as follows. In Section 2, we present an example protocol and background on protocol formalization and Isabelle/HOL. In Section 3, we explain our assumptions and modeling strategy as well as the details of our formal model. In Section 4, we present the protocols that we formalize and the proofs of their security properties. In Section 5, we survey related work and we draw conclusions in Section 6.

2. BACKGROUND

2.1 A Protocol Example

As an example of a physical proximity protocol, we present a simplified version of *authenticated ranging*, shown in Figure 1 (see [9, 12] for details on authenticated ranging).

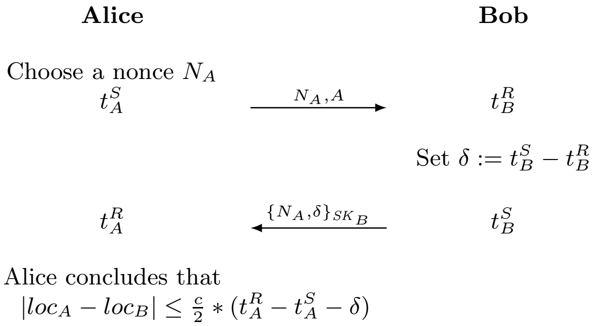

Figure 1: Simplified Authenticated Ranging

The protocol's objective is for the verifier (*Alice*) to determine a reliable upper bound on the distance to the trusted prover (*Bob*) in an adversarial environment. To achieve this, Alice uses her knowledge about the communication technology that she and Bob use to exchange information. She uses the protocol to measure the round-trip time-of-flight of a signal (traveling with speed c) between her and Bob. In particular, she creates a fresh, unguessable nonce and sends it to Bob at time (t_A^S). After receiving the nonce, Bob concatenates it with the processing time δ (the time between receiving the nonce and sending his response) and signs the message with his private key to prove that the message originates from him. Upon receiving the reply, Alice notes the time of reception t_A^R and calculates the time-of-flight, $t_A^R - t_A^S - \delta$. Since the computation time is included in the calculation of the distance, the prover (the owner of the signing key used) must be trusted. As an application for such a protocol, imagine a door-locking system that requires that a legitimate key (such as an RFID card) must be close to a door for the door's lock to open. Existing systems that can make time measurements with the necessary (nanosecond) precision for time-of-flight and processing time measurements include ultra wideband ranging systems [1].

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

2.2 The Inductive Approach

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

2.3 Isabelle/HOL

Isabelle is a generic theorem prover with a specialization for higher-order logic (HOL). We will avoid Isabelle-specific details as far as possible or explain them in context, as needed. See [29] for details on Isabelle/HOL.

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

Central to our work is the ability to define (possibly 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 proof rule for proof by induction. Examples of this and datatype definitions are provided in Section 3.

3. FORMAL MODEL

In this section, we present our model, which incorporates node location, time, and a notion of communication distance. Before presenting the technical details, we provide a high-level introduction to the concepts modeled.

3.1 Concepts Modeled

Agents. We consider a set of communicating agents, consisting of honest and dishonest agents. Honest agents follow the protocol rules, whereas dishonest agents (also called intruders) deviate arbitrarily from the protocol, seeking to subvert it. Each agent has an associated location that does not change over time, 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 messages or to analyze intercepted messages using known keys in cryptographic operations.

Network. We model the network as a matrix describing the connectivity of transmitters and receivers. The agent *Alice* can therefore send messages directly to the agent *Bob* if and only if one of Alice's transmitters is connected in the matrix to one of Bob's receivers. The matrix entries express the lower bounds on the signal propagation time from a transmitter to a receiver. They therefore formalize not only whether direct communication is possible, but also the effect of different communication technologies with different signal propagation velocities, e.g., radio and ultrasound transmission.

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

Time. Protocols such as the authenticated ranging example from Section 2.1, require a notion of time. In particular, our model must correctly describe temporal dependencies between related events, such as a send-event preceding a receive-event. Moreover, we need clocks accessible by agents for them to associate events with time-stamps. This is realized by tagging every event with the corresponding time-stamp. Temporal dependencies and clock access by agents are modeled by inductive rules that account for constant offsets of local clocks. If required, the model can also capture the stronger requirement of synchronized clocks.

Implications for the Intruder Model. In most formal approaches to security protocol analysis, the intruder is modeled as a single entity, following the Dolev-Yao intruder [17]. This intruder can defy the laws of physics by simultaneously observing all network traffic, an abstraction which is reasonable for reasoning about protocols involving properties not dependent on time or distance. Moreover, cryptography is modeled as a black box (the perfect cryptography assumption) where the intruder can construct messages in different predefined ways, but he cannot break cryptography, for example by decrypting a message without an appropriate key. In our model, we also employ the perfect cryptography assumption.

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

3.2 Agents and the Environment

We now present our model and sketch its formalization in Isabelle/HOL.

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

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

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

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

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

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

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

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

$$\mathit{cdist}_{\mathit{LoS}}(A,B) = \frac{\mid \mathit{loc}_A - \mathit{loc}_B \mid}{c},$$

where c is the speed of light. The function $\mathit{cdist}_{\mathit{LoS}}$ defines a pseudometric on the set of agents. Namely, it is non-negative, symmetric, and satisfies the triangle inequality; however, two agents can have the same location.

The value computed by $\mathit{cdist}_{\mathit{LoS}}$ only depends on the location of the agents A and B and is independent from the network topology. We model the network topology using the function $\mathit{cdist}_{\mathit{NET}} : \mathit{transmitter} \times \mathit{receiver} \to \mathbb{R}_{\geq 0} \cup \{\perp\}$, whose value depends on the communication medium used by the given transceivers, obstacles between transmitters and receivers, and other environmental factors.
$\mathit{cdist}_{\mathit{NET}}(\mathit{Tx}_A^i, \mathit{Rx}_B^j) = \perp$ denotes that Rx_B^j cannot receive transmissions from Tx_A^i. In contrast, $\mathit{cdist}_{\mathit{NET}}(\mathit{Tx}_A^i, \mathit{Rx}_B^j) =$

t, where $t \neq \bot$, describes that Rx_B^j may receive signals (messages) emitted by Tx_A^i after a delay of t time units. Since we assume that information cannot be transmitted faster than with the speed of light, we always require that

$$\mathit{cdist}_{LoS}(A, B) \leq \mathit{cdist}_{NET}(Tx_A^i, Rx_B^j).$$

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

Relation between the two Notions of Distance. The following example relates the two notions of distance: communication and physical. The left side of Figure 2 illustrates the nodes and their environment. Here, edges denote line-of-sight connections (shortest paths in Euclidian 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 does not depend on communication obstacles or physical properties of the communication medium.

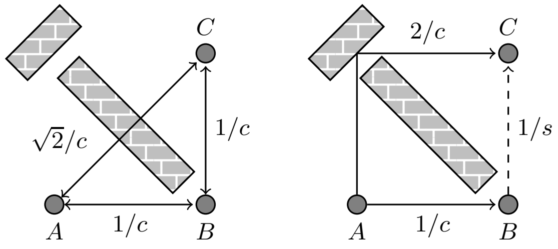

Figure 2: Physical and Network Topology.

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

3.3 Messages and Events

Messages and Message Derivation. A message is either atomic or composed. Atomic messages are agents, times, numbers, and keys. Composed messages are hashes, pairs, and encrypted messages.

```
datatype msg = Agent agent      | Time real
             | Number nat       | Nonce agent nat
             | Key key          | Hash msg
             | MPair msg msg    | Crypt key msg
```

Nonces are tagged with the name of the agent who created them and a unique identifier. This ensures that nonces created by different agents never collide. Indeed, even colluding intruders must communicate to share a nonce. Similarly, keys are assigned a unique value, whereby the set of keys is partitioned into those used for signatures, asymmetric encryption, and symmetric encryption. An inverse operator $\cdot^{-1}$ is defined for all three 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 use the notation $\{m\}_{SK}$ for the signature of the message m with the key SK.

Given a set of messages, an agent can derive new messages by decomposing and composing given messages. We formalize these message derivation capabilities with two inductively defined operators, *analz* and *synth*, each of type $\mathit{msg\ set} \ \rightarrow \mathit{msg\ set}$. The rules comprising *analz* are listed in Figure 3 and specify decomposition by message decryption and projection on pairs. The rules defining *synth*, given in Figure 4, construct messages by pairing, encryption, signing, hashing of terms, and the generation of numbers, time values, and agent names.

$$\frac{m \in M}{m \in \mathit{analz}\ M}\ \textsc{inj} \qquad \frac{\{m\}_k \in \mathit{analz}\ M \qquad k^{-1} \in \mathit{analz}\ M}{m \in \mathit{analz}\ M}\ \textsc{dec}$$

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

Figure 3: Rules for $\mathit{analz}\ \ M$

$$\frac{m \in M}{m \in \mathit{synth}\ M}\ \textsc{inj} \qquad \frac{m \in \mathit{synth}\ M \qquad n \in \mathit{synth}\ M}{(m, n) \in \mathit{synth}\ M}\ \textsc{pair}$$

$$\frac{m \in \mathit{synth}\ M \qquad k \in M}{\{m\}_k \in \mathit{synth}\ M}\ \textsc{enc} \qquad \frac{m \in \mathit{synth}\ M}{\mathit{Hash}\ m \in \mathit{synth}\ M}\ \textsc{hash}$$

$$\frac{}{\mathit{Time}\ t \in \mathit{synth}\ M}\ \textsc{time} \qquad \frac{}{\mathit{Agent}\ a \in \mathit{synth}\ M}\ \textsc{agent}$$

$$\frac{}{\mathit{Number}\ n \in \mathit{synth}\ M}\ \textsc{number}$$

Figure 4: Rules for $\mathit{synth}\ \ M$

By combining the operators *analz* and *synth* with the ability to create nonces, one may define the set of all messages that a given agent A can create from a set of messages M as $\mathit{synth}(\mathit{analz}\ \ M \cup \{\mathit{Nonce}\ A\ n \mid n \in \mathit{nat}\})$. Note that a rule for nonce creation could be added to our set of *synth* rules. However, since nonce creation is agent specific, this would also make *synth* agent specific. We prefer an agent-independent set of *synth* rules as it simplifies reasoning later.

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

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

A *trace* is a list of timed events $(t, e) \in \mathit{real} \times \mathit{event}$ with time-stamp t and event e. A timed event $(t^S, \mathit{Send}\ \mathit{Tx}_A^i\ m)$ in a trace denotes the fact that agent A has sent a message m using his transmitter Tx_A^i at time t^S. The agent A cannot designate a receiver and the *Send*-event can be received later by multiple agents, i.e. the trace may contain multiple *Recv*-events $(t^R, \mathit{Recv}\ \mathit{Rx}_B^j\ m)$ at times t^R by receivers Rx_B^j in accordance with the network topology.

A *Claim*-event models a belief or conclusion made by a protocol participant, formalized as a message. For example, after successfully completing a run of the authenticated ranging protocol (see Section 2.1) with Bob, Alice concludes 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 this condition holds for all traces containing this claim event and the protocol is used in an environment consistent with the model (as defined by *loc* and cdist_{NET}).

Knowledge and Used Messages. Each agent A possesses some initial knowledge, denoted $\mathit{initKnows}_A$. We use local contexts [5] to leave the initial knowledge unspecified and instantiate it in the context of a given protocol with the required properties. In a system run with trace tr, knowledge is defined as the union of the initial knowledge and all received messages.

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

Each agent can derive all messages in the set $\mathit{DM}_A(tr)$ by analyzing known messages, using his own nonces, and synthesizing new messages.

$$\begin{aligned}\mathit{DM}_A(tr) \equiv\ \ &\mathit{synth}(\ \mathit{analz}(\mathit{knows}_A(tr))\\ &\qquad \cup\ \{\mathit{Nonce}\ A\ n \mid n \in \mathit{nat}\})\end{aligned}$$

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

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

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

3.4 Network, Intruder, and Protocols

We now describe the inductive rules defining the set of traces $\mathit{Tr}(\mathit{proto})$ for a system parameterized by a protocol *proto*. These rules describe the network behavior, the possible actions of the intruders, and the actions taken by honest agents following the protocol rules.

Network Rule. The *Net*-rule models the message reception by receivers associated with agents. A *Recv*-event always has a preceding *Send*-event. A *Send*-event from some transmitter may result in a corresponding *Recv*-event at a receiver only if the receiver can receive messages from the transmitter as specified by cdist_{NET}. The time delay in the corresponding events is bounded below by the communication distance between transmitter and receiver.

$$\frac{\begin{array}{c} tr \in \mathit{Tr}(\mathit{proto}) \quad (t^S, \mathit{Send}\ \ \mathit{Tx}_A^i\ \ m) \in tr \\ \mathit{cdist}_{NET}(\mathit{Tx}_A^i, \mathit{Rx}_B^j) = t_{AB} \quad t_{AB} \neq \bot \\ t^R \geq \mathit{maxtime}(tr) \quad t^R \geq t^S + t_{AB} \end{array}}{tr.(t^R, \mathit{Recv}\ \ \mathit{Rx}_B^j\ \ m) \in \mathit{Tr}(\mathit{proto})}\ \textsc{Net}$$

If there is a *Send*-event in the trace and the premises of the *Net*-rule are fulfilled, a corresponding *Recv*-event can be appended to the trace. The restrictions listed earlier are ensured by $t_{AB} \neq \bot$ and $t^R \geq t^S + t_{AB}$. Here, t_{AB} is the communication distance between the receiver and the transmitter, t^S the sending time, and t^R the receiving time.

Note that a given *Send*-event can result in an unlimited number of *Recv*-events at the same receiver at different times. This is because cdist_{NET} models the minimal communication distance and messages may also arrive at later times, for example due to the reflection of the signal carrying the message. In addition, a *Send*-event can result in multiple *Recv*-events at different receivers, modeling for example broadcast communication. Finally, note that message loss or message interception by the intruders is captured by never applying the *Net*-rule for a given *Send*-event in a trace, even if all premises are fulfilled.

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

Our rules specify that time-stamps increase monotonically in every trace. This ensures that the partial order on timed events induced by the time-stamps (events can happen simultaneously) is consistent with the order of events in the list. We use lists instead of sets of events since the total order on events (induced by the list structure) simplifies proofs in some cases. Additionally, the temporal order of events does not capture all causal dependencies. For example, the *Claim*-event caused by the reception of a message can have the same time-stamp as the *Recv*-event in our model.

Intruder Rule. The *Fake*-rule describes the intruders' behavior. Namely, intruders can always send any message derivable from their knowledge.

$$\frac{tr \in \mathit{Tr}(\mathit{proto}) \quad m \in \mathit{DM}_{I_A}(tr) \quad t \geq \mathit{maxtime}(tr)}{tr.(t, \mathit{Send}(\mathit{Tx}_{I_A}^k, m)) \in \mathit{Tr}(\mathit{proto})}\ \textsc{Fake}$$

Since knowledge is distributed, we use explicit *Send*- and *Recv*-events to model the exchange of information between colluding intruders. For example, if one intruder wishes to share an intercepted message with another intruder, he must do so over the network. Therefore the lower bounds on transmission delays also hold for the intruders. However, with an appropriate $cdist_{NET}$ function, it is possible to model an environment where the intruders have a high-speed network, for example, for carrying out worm-hole attacks. Restrictions on the degree of cooperation between intruders can be modeled as predicates on traces, e.g., an intruder never sends his long-term secrets. Such predicates would then be added to the assumptions in the proofs of a security property.

Protocols. A protocol is described by a set of protocol steps. Each step corresponds to the behavior of an agent participating in a protocol run. A protocol step is a function of type $agent \times trace \to (time \times pevent \times msg)\ set$ returning triples that describe the possible actions an agent can take given a trace. The value of type *pevent* describes the type of action to be taken by the agent. Therefore a *pevent* describes either a *Send*-event with a given transmitter id or a *Claim*-event.

$$\textbf{datatype}\ pevent\ =\ \mathsf{SendEv}\ nat \mid \mathsf{ClaimEv}$$

The *createEv* function translates the triples returned by a step function to events.

$$createEv(A, \mathsf{SendEv}\ \ k, m) \equiv Send\ \ Tx_A^k\ \ m$$
$$createEv(A, \mathsf{ClaimEv}\ , m) \equiv Claim\ A\ m$$

The actions of an agent A should only depend on his own previous actions and observations. We therefore define A's view of a trace tr as the projection of tr to those events involving A. Since the timestamps of events refer to the global clock, the *view* function must account for the offset of A's clock.

$$view\ A\ tr \equiv [(t + coffset(A), ev) \mid (t, ev) \in tr \wedge occursAt(ev) = A]$$

The uninterpreted function $coffset : agent \to real$ translates global to local times and returns the offset of an agent's clock with respect to the global clock. The function $occursAt : event \to agent$ returns the agent associated with a given event and is defined as follows.

$$occursAt(\mathsf{Send}\ (Tx_A^i)\ m) \equiv A$$
$$occursAt(\mathsf{Recv}\ (Rx_A^i)\ m) \equiv A$$
$$occursAt(\mathsf{Claim}\ A\ m) \equiv A$$

Using these definitions, the protocol rule *Proto* applies all step functions to all agents and their views of valid traces, creates the events corresponding to the values returned by the step functions, and appends the events to the traces. The rule accounts for clock offsets and translates the local times of events returned by the step function back to the global times used in the trace.

$$\frac{\begin{array}{c} tr \in Tr(proto) \quad step \in proto \\ (t, pEv, m) \in step(view(H_A, tr), H_A) \\ m \in DM_{H_A}(tr) \quad t' = t - coffset(H_A) \\ t' \geq maxtime(tr) \end{array}}{tr.(t', createEv(H_A, pEv, m)) \in Tr(proto)}\ \textsc{Proto}$$

The restriction that all messages must be in $DM_{H_A}(tr)$ ensures that agents only send messages derivable from their knowledge. This will be the case for all protocols of interest, since honest agents running the protocol should be able to derive all protocol messages, otherwise some initial knowledge is missing or the protocol is not executable.

The above definition allows us to reason about all protocols, or subsets thereof defined by a predicate, for example, the set of all protocols where agents do not send their long term secrets. To reason about properties of a concrete protocol, it is more convenient to use a non-parameterized definition where every protocol step is directly modeled as an inductive rule. We therefore use the parameterized form in order to reason about properties of an entire class of protocols, whereas we use the non-parameterized form to reason about properties of an individual protocol. Proving that the sets of traces defined by the two inductive definitions for a given protocol coincide is usually easy. Therefore we can use facts proved about a class of protocols in proofs about an individual protocol given by a non-parameterized definition with the corresponding induction principle. We show, for example, that for all protocols not sending long-term secrets, messages containing signatures are always first sent by the owner of the key; afterwards we use this fact in the proof of an authenticated ranging protocol.

4. APPLYING THE MODEL

In this section, we apply our model to the verification of three protocols: (simplified) authenticated ranging, ultrasonic distance-bounding, and TESLA broadcast authentication. Each protocol uses cryptographic primitives as well as physical characteristics of the communication technology, environment, or network topology, in order to provide security guarantees.

4.1 Authenticated Ranging

To define the trace set for the authenticated ranging protocol introduced in Section 2.1, we add protocol-specific rules to the protocol-independent *Fake*- and *Net*-rules presented in Section 3. Following the protocol description in Figure 1, we add the rules below:

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

$$\frac{\begin{array}{c} tr \in TR \quad N_A \notin used(tr) \\ t \geq maxtime(tr) \end{array}}{tr.(t, Send\ \ Tx_A^r\ \ N_A) \in TR}\ \text{AR1}$$

2. The reply rule (*AR2*) allows receivers of an initial message to continue the protocol by responding accordingly.

$$\frac{\begin{array}{c} tr \in TR \quad (t^R, Recv\ \ Rx_B^r\ \ N_A) \in tr \\ t^S \geq maxtime(tr) \end{array}}{tr.(t^S, Send\ \ Tx_B^r\ \{N_A, t_S - t_R\}_{SK_B}) \in TR}\ \text{AR2}$$

3. The final rule (*AR3*) introduces a *Claim*-event and models the conclusion of an initiator A who has re-

ceived a response to his initial challenge.

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

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

For this protocol, we define the initial knowledge of each agent A as the signing key SK_A and the public signature keys PK_B, for all agents B.

Security Analysis. As explained in Section 2.1, the protocol is intended to compute a reliable upper bound on the physical distance between honest agents executing the protocol.

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

To prove this theorem, we first establish three lemmas about the ordering of messages. The first lemma gives a lower bound on the time between an agent first using a nonce and another agent using the same nonce later.

Lemma 4.2. *Let A be an agent and let $(t_A^S, \mathit{Send}\ \ Tx_A^i\ \ m_A)$ be the first event in the trace tr containing nonce N. If there is another event $(t_B^S, \mathit{Send}\ \ Tx_B^j\ \ m_B) \in tr$ with $A \neq B$ such that m_B contains N then $t_B^S - t_A^S \geq \mathit{cdist}_{LoS}(A, B)$ holds.*

The next lemma is similar, but concerns the earliest possible time that an honest agent can receive a nonce. We later use it to bound the processing time contained in the second protocol message.

Lemma 4.3. *Let A be an honest agent and let $(t_A^S, \mathit{Send}\ \ Tx_A^i\ \ m_A)$ be the first event in trace tr containing nonce N in m_A. If tr contains an event $(t_B^R, \mathit{Recv}\ \ Rx_B^j\ \ m_B)$ where m_B contains N, then $t_B^R - t_A^S \geq \mathit{cdist}_{LoS}(A, B)$ holds.*

Our final lemma concerns signatures and their creation time.

Lemma 4.4. *Let A be an honest agent and let $(t_B^S, \mathit{Send}\ \ Tx_B^i\ \ m_B) \in tr$ be an event in trace tr where the message contains a signature of A. Then there is a Send-event $(t_A^S, \mathit{Send}\ \ Tx_A^j\ \ m_A) \in tr$ with m_A containing the same signature and $t_B^S - t_A^S \geq \mathit{cdist}_{LoS}(A, B)$.*

We are now ready to prove Theorem 4.1. In addition to these lemmas, we need the fact that for an honest agent B, δ sent in the second protocol message always corresponds to the time between receiving the first message and sending the third message, i.e., honest agents follow the protocol rules.

Proof. To prove Theorem 4.1 we proceed as follows. Since only rule *AR3* adds events of the form $(t_A^C, \mathit{Claim}\ A\ (B, d))$, we know from the premises of *AR3* that N_A originates at event $(t_A^S, \mathit{Send}\ \ Tx_A^r\ \ N_A)$ in the trace. Furthermore, we know that there is an event $(t_A^R, \mathit{Recv}\ \ Rx_A^r\ \{N_A, \delta\}_{SK_A})$, where $d = \frac{c}{2} * (t_A^R - t_A^S - \delta)$.
From the above, there must be a corresponding *Send*-event in the trace at time t_C^S, with $t_A^R - t_C^S \geq \mathit{cdist}_{LoS}(C, A)$, produced by some agent (possibly an intruder). Using Lemma 4.4, we conclude that B sent a message containing the signature at time t_B^S, where $t_C^S - t_B^S \geq \mathit{cdist}_{LoS}(B, C)$. This message must result from an application of rule *AR2*, since B is assumed honest. Hence there is a *Recv*-event at time t_B^R and $\delta = t_B^S - t_B^R$. Finally we use Lemma 4.3 to show that $t_B^R - t_A^S \geq \mathit{cdist}_{LoS}(A, B)$ and sum up the inequalities.

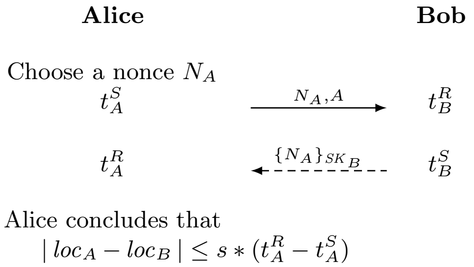

Figure 5: Distance Bounding Protocol

$$\begin{array}{ll} & t_A^R - t_A^S - \delta \\ = & t_A^R - t_C^S + t_C^S - t_B^S + t_B^R - t_A^S \\ \geq & \mathit{cdist}_{LoS}(C, A) + \mathit{cdist}_{LoS}(B, C) + \mathit{cdist}_{LoS}(A, B) \\ \geq & 2 * \mathit{cdist}_{LoS}(A, B) \end{array}$$

Therefore we conclude that

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

□

4.2 Ultrasound Distance Bounding

Our second example is a simplified protocol for *distance bounding* using ultrasound. The goal of the protocol in Figure 5 is for the initiator Alice to determine a reliable upper bound on the distance to the responder Bob. The dashed arrow denotes message transmission using ultrasound and s denotes ultrasound's propagation speed. Using the speed of ultrasound to measure distance has the advantage that due to its slowness (several orders of magnitude slower than the propagation speed of radio signals), we can safely neglect the transmission time of the first message and the time for signing messages.

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

$$\begin{array}{l} \mathit{cdist}_{NET}(Tx_A^i, Rx_B^{us}) \neq \bot \\ \Rightarrow \mathit{cdist}_{NET}(Tx_A^i, Rx_B^{us}) \geq \dfrac{|\,loc_A - loc_B\,|}{s} \end{array}$$

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

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

We now give the inductive rules for the protocol.

1. The start rule (*DB1*) initiates a protocol run.
$$\frac{tr \in TR \quad N_A \notin used(tr) \quad t \geq maxtime(tr)}{tr.(t, Send\ \ Tx_A^r\ \ N_A) \in TR}\ \text{DB1}$$
2. The reply rule (*DB2*) allows receivers of initial messages to respond according to the protocol rules.
$$\frac{tr \in TR \quad (t_R, Recv\ \ Rx_B^r\ \ N_A) \in tr \quad t_S \geq maxtime(tr)}{tr.(t_S, Send\ \ Tx_B^{us}\ \{N_A\}_{SK_B}) \in TR}\ \text{DB2}$$
3. The final rule (*DB3*) introduces a *Claim*-event when an initiator A receives a response to his initial challenge.
$$\frac{tr \in TR \quad (t_A^S, Send\ \ Tx_A^r\ \ N_A) \in tr \quad (t_A^R, Recv\ \ Tx_A^{us}\ \{N_A\}_{SK_B}) \in tr \quad t_A^R \geq maxtime(tr)}{tr.(t_A^R, Claim\ A\ (B, (t_A^R - t_A^S) * s)) \in TR}\ \text{DB3}$$
This models what A concludes about a signal that (apparently) traveled back-and-forth between A and B in the time $t_A^R - t_A^S$, namely that $s * (t_A^R - t_A^S)$ is a reliable upper bound on the distance between A and B.

Security Analysis. The security property of the distance bounding protocol is similar to the authenticated ranging example. But since the computation time of the prover is not used in computing the distance, the protocol does not require an honest prover.

THEOREM 4.5. *Let A be an honest agent and B be any agent. Furthermore consider a valid trace tr, where $(t, Claim\ A\ (B, d)) \in tr$. Then $d \geq |\, loc_A - loc_B \,|$.*

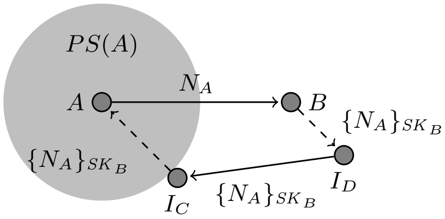

Figure 6: Attack on DB using Ultrasound

The protocol is insecure without additional assumptions. There is an attack involving two colluding intruders illustrated in Figure 6. Here I_D is close to B, receives the reply over ultrasound, and forwards it to another intruder I_C close to A using radio. Then I_C delivers the message to A over ultrasound. We have proven (in Isabelle/HOL) that this attack is captured in our model. The scenario is depicted in Figure 6, where $PS(A)$ denotes the private space of A. $PS(A)$ is the largest circle centered at A where A can ensure that no intruder is inside the circle. The inequality involving the communication distances necessary for such an attack to work is

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

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

Therefore we prove the security of the protocol under an additional assumption. The verifier A ensures that the prover B is in his private space. This same assumption is used in a number of protocols (e.g., [36, 11]) for location-based access control and device pairing. This assumption is captured by the inequality

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

We now prove Theorem 4.5 under the additional assumption that B is in the private space of A.

PROOF. We prove this by induction over traces using Lemma 4.2, which holds for all protocols. Our proof uses the parameterized protocol definition and we have shown that the set of traces defined by a parameterized definition of this distance bounding protocol corresponds to the version with separate protocol rules presented above. In our Isabelle/HOL formalization, we have only proved the general lemmas once and reused them in the security proofs of both protocols.

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

Since there must be a *Send*-event corresponding to the *Recv*-event with the signature of B, the sender is either B or an intruder I. In the first case, the *Send* occurs at time t_B^S, with $t_A^R - t_B^S \geq cdist_{NET}(Tx_B^{us}, Rx_A^{us})$. From Lemma 4.2 it follows that $t_B^S \geq t_A^S$, since N_A is included in the message. Together with the previous inequality and the assumption that messages received by ultrasound receivers do not travel faster than s, we conclude that $t_A^R - t_A^S \geq cdist_{NET}(Tx_B^{us}, Rx_A^{us}) \geq |\, loc_A - loc_B \,|/s$.

Finally, consider the case where the message is sent by the intruder I at time t_I^S. Using the assumption that B is in the private space of A, it follows that the distance between A and I is at least the distance between A and B. Additionally, the assumptions state that a message received by Rx_A^{us} has not travelled faster than s. Together with $t_I^S \geq t_A^S$, which follows from Lemma 4.2, this completes the proof.

$$\begin{aligned} t^R_A - t^S_A &\geq t^R_A - t^I_A \\ &\geq cdist_{NET}(Tx^j_I, Rx^{us}_A) \\ &\geq \mid loc_A - loc_I \mid /s \\ &\geq \mid loc_A - loc_B \mid /s \end{aligned}$$

□

Note that the proof does not use the fact that the second protocol message is authenticated by B (which is necessary for Lemma 4.4). Authentication is guaranteed by the fact that A knows that B is in its private space. Therefore even a simplified version of the protocol, where the second message is replaced with the pair (N_A, B), would be secure under the private space assumption.

4.3 A Delayed Key Disclosure Protocol

In our final example, we model and verify a *Delayed Key Disclosure* protocol used for broadcast authentication in resource constrained environments (such as sensor networks), where asymmetric cryptography is not available. A suite of such protocols is described in [32]. We formalize the TESLA broadcast authentication protocol developed by Perrig et al. [31].

In TESLA, the sender commits to a set of keys $(K_i)_{1 \leq i \leq n}$, which are used in keyed MACs. These keys 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. The key $K_i = Hash^{n-i}(H_0)$ is then used within its validity window $[T_{i+1}, T_{i+2}[$, where $T_i = T_0 + i * valwin$, to generate a MAC for the messages sent in the same window. After K_i's validity window has passed, the sender releases the key K_i corresponding to the release schedule. We use $[T_{i+1}, T_{i+2}[$ as K_i's release window, which corresponds to the release schedule of TESLA described in [31]. Figure 7 depicts the above.

In our formalization, we use the abbreviation $MAC_{K_i}(m) = (Hash(K_i, m), m)$ as the keyed MAC containing the message m. The secret H_0 is only contained in BR's initial knowledge, every other agent's initial knowledge only contains H_n.

The protocol rules are formalized as follows:

1. The rule (*DKD-BR*) formalizes the behavior of the broadcast source. According to the release schedule, Br chooses the currently valid key (K_i in time-interval i) and authenticates the message m. He also releases the old key (K_{i-2}) valid in interval $i - 2$.

$$\frac{tr \in TR \quad t \in [T_{i-1}, T_i[\\ t \geq maxtime(tr)}{tr.(t, Send\ \ Tx_{Br}\ (MAC_{K_i}(m), K_{i-2})) \in TR} \text{ DKD-BR}$$

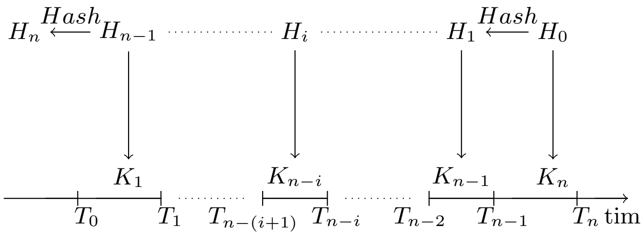

Figure 7: Association of Hash Chain Elements to Time-Slots

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

$$\frac{tr \in TR \quad t^{R1} < T_i \quad i \leq n \\ (t^{R1}, Recv\ \ Rx_R\ (MAC_{K_i}(m), K_{i-2})) \in tr \\ (t^{R2}, Recv\ \ Rx_R\ (MAC_{K_{i+2}}(m'), K_i)) \in tr \\ t \geq maxtime(tr)}{tr.(t, Claim\ R\ (m, i)) \in TR} \text{ DKD-CO}$$

Note that in the second rule's premises, we do not restrict the arrival time of the released key; it just has to arrive once. The premises could be further weakened by requiring only the reception of a later key (K_j, where $j > i$) allowing verification of all earlier keys, even if the messages disclosing these have been lost.

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

THEOREM 4.6. *Let tr be a valid trace. If* $(t^C, Claim\ H_R\ (m, i)) \in tr$*, then there exists* $(t^S, Send\ \ Tx_{Br}\ (MAC_{K_i}(m), K_{i-2})) \in tr$*, where* $t^S \in [T_{i-1}, T_i[$ *holds.*

We have proved Theorem 4.6 using the following lemmas:

LEMMA 4.7. *Suppose that* $0 \leq l \leq n$*, A is an agent other than Br, and tr is a valid trace. If* $K_l \sqsubseteq DM_A(tr)$ *(i.e., agent A can derive a message from his observations of the trace tr that contains* K_l*), or if* $(t, Send\ \ Tx_A\ \ X) \in tr$*, where* $K_l \sqsubseteq X$*, then* $maxtime(tr) \geq T_{l+1}$*.*

This lemma states that nobody can use a key before it has been released by the broadcast source (according to the release-schedule). The proof can be found in Appendix A.

LEMMA 4.8. *Suppose that tr is a valid trace and that* $(t, Send\ \ Tx_A\ \ Y) \in tr$*, where* $Hash(K_l, m) \sqsubseteq Y$*. Furthermore suppose that* $maxtime(tr) < T_{l+1}$*, and* $0 < k < n$*, then there exists an event* $(\tilde{t}, Send\ \ Tx_{Br}\ (MAC_{K_l}(m), K_{l-2})) \in tr$ *with* $\tilde{t} \in [T_{l-1}, T_l[$*.*

In this last lemma, we claim that if an arbitrary agent A sends a message m authenticated with a key K_l that has not yet been released, then the honest broadcaster Br must have sent the corresponding MAC at the right time.

Proof. We only have to consider the *Fake* rule and the case where the event $(t_I, Send\ \ Tx_I^k\ X)$, with $Hash(K_l, m) \sqsubseteq X$, is added to the trace tr with $maxtime(tr) < T_{l+1}$. $Hash(K_l, m) \sqsubseteq DM_I(tr)$ follows from the premises of the rule. This implies that one of the following is true: (i) I received a message containing K_j for some $j \geq l$ or (ii) I received a message containing $Hash(K_l, m)$. We show that case (i) is impossible using lemma 4.7 since $maxtime(tr) \geq T_{l+1}$ contradicts $maxtime(tr) < T_{l+1}$. The proof for case (ii) uses the fact that there is a corresponding *Send*-event and then applies the induction hypothesis. □

The proof of Theorem 4.6 using the previous lemma is straightforward.

Proof. Since only *DKD-CO* adds events of the form (t, *Claim* H_R (m, i)), we need not consider the other rules. From the premises of *DKD-CO*, we conclude that there is a *Recv*-event with message $(MAC_{K_i}(m), K_{i-2})$ and time t^{R1}, where $t^{R1} \in [T_{i-1}, T_i[$. Therefore, there must be a corresponding *Send*-event *sev* for the message with $t^S < T_i$. We now consider the prefix of the trace up to *sev*. Since *sev* is the last event in the trace, $maxtime(tr) < T_{i+1}$ holds and with the premises from *DKD-CO*, we can apply lemma 4.8 which completes the proof. □

In summary, the combination of a timed communication and message theory in our model allowed us to verifiy TESLA, which uses time and properties of hash functions in a nontrivial way to achieve broadcast authentication. The assumption about synchronized local clocks used in the proof can be ommitted by using the clock offset introduced in Section 3.4, leading to possibly weaker security guarantees.

5. 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 [4, 15, 8], and interactive methods, such as theorem proving [30]. In both settings, it is standard to formalize an intruder model based on the Dolev-Yao model, which identifies the intruder with the network.

We now summarize formal approaches that address aspects of time, network topology, and location, which are the three central notions captured by our model. Most approaches formalizing time only focus on time-stamps, which are used to reason about key-expiration (e.g., in protocols like Kerberos. The models of [6, 16, 19] are based on discrete time, whereas [38] uses dense time. Corin et al. use timed automata [3] to model timing attacks and timing issues like timeouts and retransmissions in security protocols [14]. In [21] the authors use a real-time process algebra to model and analyze μ-TESLA. The protocol is proved to achieve a time-dependent form of integrity for a set of messages sent by the broadcast source, abstracting away from the network and the topology.

Network topology has been considered in formal approaches for analyzing routing protocols in ad hoc networks [2, 28, 41]. Here, only the ability of a node to receive a signal sent by another node is considered and communication distance is neglected.

Node location has been, to our knowledge, only used in informal proofs. For example, Sastry et al. [36] propose a protocol for verifying location claims based on ultrasonic communication and provide an informal proof of its security and reliability. Since the protocol does not provide location discovery, authentication is neither considered in the protocol nor in the proof. Other approaches only formalize the related notion of relative distance. In Meadows et al. [26], an authentication logic is extended to handle relative distance and is used to prove the security of a newly proposed distance bounding protocol. Here, the distance between two nodes is axiomatically defined as the minimal time-of-flight of a message from the verifier to the prover and back. Different signal propagation speeds are not captured in the model. A formal comparison of the two approaches would be interesting and a possible subject of future work.

6. CONCLUSION

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

Isabelle Formalization. Our model and all the proofs presented in this work have been formalized and checked in Isabelle/HOL. The proofs are available at [7]. Our Isabelle formalization makes use of existing theories for real vector spaces including the formalization of the Cauchy-Schwarz inequality [33]. While Isar's support for reasoning with chains of inequations was very useful, more automation support for reasoning about linear formulas over the reals is still desirable. We use local contexts [5] to transfer general results to the concrete protocol setting using interpretation. So most of the theory comprises general results applicable to arbitrary protocols and the security proofs of the protocols are comparable small. The complete formalization comprises 136 pages of PDF documentation (6820 lines) with the examples taking 13 pages (578 lines), 15 pages (725 lines), and 22 pages (1029 lines) respectively.

Future Work. As future work, we plan to extend our model to capture additional properties of wireless security protocols. These protocols include mutual distance bounding [10], secure time synchronization [39, 20, 25], and secure localization [36, 34]. We intend to refine our model to capture message sizes and transmission rate, rapid bit exchange, and online guessing attacks. Finally, we intend to work toward the development of fully automatic proof tools that can analyze the security properties of such protocols.

7. REFERENCES

[1] Multispectral Solutions, Inc. http://www.multispectral.com/.

[2] G. Acs, L. Buttyan, and I. Vajda. Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2006.

[3] R. Alur and D. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 1994.

[4] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, et al. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. *Computer Aided Verification, CAV*, 2005.

[5] C. Ballarin. Interpretation of locales in Isabelle: Theories and proof contexts. *Mathematical Knowledge Management*, 2006.

[6] G. Bella. *Formal Correctness of Security Protocols.* 2007.

[7] Benedikt Schmidt and Patrick Schaller. Isabelle Theory Files. http://people.inf.ethz.ch/benschmi/ProtoVerPhy/.

[8] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *CSFW*, 2001.

[9] S. Brands and D. Chaum. Distance-bounding protocols. In *EUROCRYPT '93*, 1994.

[10] S. Capkun, L. Buttyan, and J.-P. Hubaux. Sector: secure tracking of node encounters in multi-hop wireless networks. In *SASN '03: ACM Workshop on Security of of ad hoc and sensor networks*, 2003.

[11] S. Capkun and M. Cagalj. Integrity regions: authentication through presence in wireless networks. In *WiSe '06: ACM workshop on Wireless security*, 2006.

[12] S. Capkun and J. Hubaux. Secure positioning in wireless networks. *Selected Areas in Communications, IEEE Journal on*, 2006.

[13] J. Clulow, G. Hancke, M. Kuhn, and T. Moore. So Near and Yet So Far: Distance-Bounding Attacks in Wireless Networks. *European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks, Hamburg*, 2006.

[14] R. Corin, S. Etalle, P. Hartel, and A. Mader. Timed analysis of security protocols. *Journal of Computer Security*, 2007.

[15] C. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In *CAV*, 2008. To appear.

[16] G. Delzanno and P. Ganty. Automatic Verification of Time Sensitive Cryptographic Protocols. *Tools and Algorithms for the Construction and Analysis of Systems: TACAS, Held as Part of ETAPS*, 2004.

[17] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE, Transactions on Information Theory*, 1983.

[18] S. Drimer and S. J. Murdoch. Keep your enemies close: Distance bounding against smartcard relay attacks. In *USENIX Security Symposium*, 2007.

[19] N. Evans and S. Schneider. Analysing Time Dependent Security Properties in CSP Using PVS. *ESORICS 2000: European Symposium on Research in Computer Security*, 2000.

[20] S. Ganeriwal, S. Čapkun, C.-C. Han, and M. B. Srivastava. Secure time synchronization service for sensor networks. In *WiSe '05: ACM workshop on Wireless security*, 2005.

[21] R. Gorrieri, F. Martinelli, M. Petrocchi, and A. Vaccarelli. Formal anaylsis of some timed security properties in wireless protocols. In *FMOODS*, 2003.

[22] G. P. Hancke and M. G. Kuhn. An rfid distance bounding protocol. In *SECURECOMM '05: Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005.

[23] M. Kuhn. An Asymmetric Security Mechanism for Navigation Signals. *Information Hiding: IH*, 2004.

[24] L. Lazos, R. Poovendran, and S. Capkun. ROPE: robust position estimation in wireless sensor networks. *Information Processing in Sensor Networks: IPSN*, 2005.

[25] M. Manzo, T. Roosta, and S. Sastry. Time synchronization attacks in sensor networks. In *SASN '05: ACM workshop on Security of ad hoc and sensor networks*, 2005.

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

[27] J. Munilla, A. Ortiz, and A. Peinado. Distance bounding protocols with void-challenges for RFID. Printed handout at the Workshop on RFID Security – RFIDSec 06, 2006.

[28] S. Nanz and C. Hankin. A framework for security analysis of mobile wireless networks. *Theor. Comput. Sci.*, 2006.

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

[30] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 1998.

[31] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The tesla broadcast authentication protocol. *RSA CryptoBytes*, 2002.

[32] A. Perrig and J. D. Tygar. *Secure Broadcast Communication in Wired and Wireless Networks.* 2002.

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

[34] K. B. Rasmussen, S. Capkun, and M. Cagalj. Secnav: secure broadcast localization and time synchronization in wireless networks. In *MobiCom '07: ACM international conference on Mobile computing and networking*, 2007.

[35] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji. Detecting relay attacks with timing-based protocols. In *ASIACCS '07: ACM symposium on Information, computer and communications security*, 2007.

[36] N. Sastry, U. Shankar, and D. Wagner. Secure

verification of location claims. In *WiSe '03: ACM workshop on Wireless security*, 2003.

[37] P. Schaller, S. Capkun, and D. Basin. Bap: Broadcast authentication using cryptographic puzzles. *International Conference on Applied Cryptography and Network Security (ACNS)*, 2007.

[38] R. Sharp and M. Hansen. Timed Traces and Strand Spaces. *16th Nordic Workshop on Programming Theory*, 2004.

[39] K. Sun, P. Ning, and C. Wang. TinySeRSync: secure and resilient time synchronization in wireless sensor networks. *ACM conference on Computer and communications security*, 2006.

[40] N. Tippenhauer and S. Čapkun. UWB-based Secure Ranging and Localization. Technical report, ETH Zurich, 2008.

[41] S. Yang and J. S. Baras. Modeling vulnerabilities of ad hoc routing protocols. In *SASN '03: ACM workshop on Security of ad hoc and sensor networks*, 2003.

APPENDIX

A. PROOFS

A.1 Authenticated Ranging

LEMMA A.1. *Let A be an agent and let $(t_A^S, Send\ \ Tx_A^i\ \ m_A)$ be the first event in the trace tr containing nonce N. If there is another event $(t_B^S, Send\ \ Tx_B^j\ \ m_B) \in tr$ with $A \neq B$ such that m_B contains N, then $t_B^S - t_A^S \geq cdist_{LoS}(A, B)$.*

PROOF. We prove this by induction on traces. The proof for the rules *Nil*, *Net*, and *AR3* follows trivially from the induction hypothesis since these rules do not add any *Send*-events. We now consider the three remaining rules.

AR1: The event $(t, Send\ \ Tx_A^r\ \ N_A)$ is appended to the trace. Since N_A is fresh, no other message containing N_A can be included in the trace.

AR2: The event $(t_B^S, Send\ \ Tx_B^r\ \ \{N_A, \delta\}_{SK_B})$ is appended to the trace and $(t_B^S - \delta, Recv\ \ Rx_B^r\ \ N_A)$ is already in the trace as required by the premises. Since every *Recv*-event is preceded by a corresponding *Send*-event, there must be a $(t^C, Send\ \ Tx_C^k\ \ N_A))$ event in the trace where $t^C \leq t_B^S - cdist_{NET}(C, B)$. From the induction hypothesis, it follows that $t^C - t_A^S \geq cdist_{LoS}(A, C)$. Using the triangle inequality for the physical distance and the consistency condition forbidding faster-than-light communication, $t_B^S - t_A^S \geq cdist_{LoS}(A, B)$ immediately follows.

Fake: The event $(t_I, Send\ \ Tx_I^k\ \ m)$ is added to the trace. If m contains a fresh nonce, there is no earlier *Send*-event containing this nonce. If m contains a nonce N created earlier by another agent, I must have received a message containing N. Therefore there must be an earlier *Send*-event containing N as a message. Using the induction hypothesis and the triangle inequality, the claim follows analogously to the *AR2* case.

□

LEMMA A.2. *Let A be an honest agent and let $(t_A^S, Send\ \ Tx_A^i\ \ m_A)$ be the first event in trace tr containing nonce N in m_A. If tr contains an event $(t_B^R, Recv\ \ Rx_B^j\ \ m_B)$ where m_B contains N, then $t_B^R - t_A^S \geq cdist_{LoS}(A, B)$ holds.*

PROOF. We prove this by induction on traces. Here, only the *Net*-rule is of interest since the proof that the lemma still holds after adding *Claim*-events and *Send*-events is trivial. The *Net*-rule adds a *Recv*-event $(t_B^R, Recv\ \ Rx_B^k\ \ m)$, which must always be preceded by a corresponding *Send*-event $(t_C^S, Send\ \ Rx_C^l\ \ m)$, where $t_C^S \leq t_B^R - cdist_{NET}(D, C)$. Using lemma A.1, we know that $t_D^S - t_A^S \geq cdist_{LoS}(A, D)$ for the event that generates the nonce at time t_A^S. Combining the two inequalities and using the triangle inequality and the consistency condition for $cdist_{NET}$, we get the desired inequality $t_B^R - t_A^S \geq cdist_{LoS}(A, B)$. □

LEMMA A.3. *Let A be an honest agent and let $(t_B^S, Send\ \ Tx_B^i\ \ m_B) \in tr$ be an event in trace tr where the message contains a signature of A. Then there is a send event $(t_A^S, Send\ \ Tx_A^j\ \ m_A)) \in tr$ with m_A containing the same signature and $t_B^S - t_A^S \geq cdist_{LoS}(A, B)$.*

PROOF. The proof steps correspond to the proof of lemma A.1. The proof additionally uses that fact that intruders cannot create signatures on behalf of honest agents since the (signing) keys of honest agents are never leaked. □

A.2 Delayed Key Disclosure

LEMMA A.4. *Suppose that $0 \leq l \leq n$, A is an agent other than Br, and tr is a valid trace. If $K_l \sqsubseteq DM_A(tr)$ (i.e., agent A can derive a message from his observations of the trace tr that contains K_l), or if $(t, Send\ \ Tx_A\ \ X) \in tr$, where $K_l \sqsubseteq X$, then $maxtime(tr) \geq T_{l+1}$.*

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

Fake: The event $(t_I, Send\ \ Tx_I^k\ \ X)$ is added to the trace tr. We only have to consider the case where $K_l \sqsubseteq X$. Here, $K_l \sqsubseteq DM_A(tr)$ follows from the premises of the rule and we can apply the induction hypothesis.

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

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

□