Assumption-Based Analysis of Distance-Bounding Protocols with CPSA

Paul D. Rowe, Joshua D. Guttman, and John D. Ramsdell

The MITRE Corporation
{prowe,guttman,ramsdell}@mitre.org

Abstract. This paper, dedicated to Andre Scedrov, was inspired by conversations with him about the physical properties of distributed systems. We use CPSA, the strand space protocol analysis tool, to analyze and classify distance-bounding protocols. We introduce a model of strand spaces that explicitly accounts for physical properties like distance. We prove that non-metric, causal facts allow us to infer distance bounds. Moreover, CPSA already provides these causal conclusions about protocols. We apply this method to numerous protocols from the literature. By taking an assumption-based perspective—rather than an attack-based perspective—we introduce a taxonomy of distance-bounding protocols that compares the relative strength of different designs.

1 Introduction

A *distance-bounding protocol* is an exchange of messages between parties that include a *prover* and a *verifier* [6,16]. The verifier wants to determine whether the prover is nearby, i.e. within some application-relevant radius. This requires authenticating the prover to some extent, since generally one wants to know which party is within the radius. For instance, if a credit card is the device acting as prover, the verifier definitely needs to know what number is associated with it so that the right number will be billed.

Distance-bounding protocols have often been weak, sometimes quixotically weak. In this paper, we will approach distance-bounding protocols in three steps.

First, although the goals of a distance-bounding protocol are essentially metric—they are about how far the prover is from the verifier—we extract a non-metric model from them, using strand spaces. From this non-metric model, together with purely local metric assertions about the time elapsed for a single participant, metric consequences about space and time will follow. Lemma 2 justifies this step back to a conclusion about the distance to the prover.

Second, we show how to use the strand space protocol analyzer CPSA to extract a set of non-metric executions for each distance-bounding protocol. From these non-metric executions, we can draw conclusions about whether a protocol achieves its metric goals, with the backing of Lemma 2.

Finally, we exhibit a taxonomy classifying distance-bounding protocols by the assumptions that they require, to be sure of achieving their goals.

Strand spaces in spacetime. The idea for strand spaces came from an analogy to spacetime diagrams in physics. A spacetime diagram organizes some physical interactions by considering the *world-lines* of some entities as they progress through time, moving in space. Moreover, the entities interact through *messages*, whether transmitted as light or as other waves or particles; these messages travel no faster than the speed of light c.

Protocol analysis is structurally similar: the world-line of a principal includes message transmissions and receptions. If a principal is *regular*, i.e. acting in accordance with the protocol under study, these transmission and reception events partition into a number of *regular strands*, meaning a finite sequence of transmission and reception events $\circ \Rightarrow \circ \Rightarrow \cdots$ permitted by some protocol role. For uniformity, we divide the actions of a Dolev-Yao adversary [12] into a collection of finite sequences of transmission and reception events; these are *adversary strands*. A protocol execution consists of a finite collection of regular and adversary strands, or initial segments of them, with two main properties:

- If a reception event receives a message m, then some transmission event must have sent m, i.e. $\circ \stackrel{m}{\rightarrow} \circ$; and
- the finite directed graph G must be acyclic, where G's nodes $\circ$ are the events, and G's arcs $\rightarrow, \Rightarrow$ are either message communications $\rightarrow$ or the succession relation between two events along the same strand $\Rightarrow$.

These are natural properties of causality. The first says that message reception needs to be causally explained by some transmission. The second is the familiar principle that causality is well-founded: You cannot go back and encourage your grandparents to beget your parents, or not to. It certainly applies in our context, in which message transmissions and receptions occur at discrete, well-separated times, and where moreover none of the activities will stretch over long (or cosmological) timescales.

Diagrams with these two properties are *bundles*, and bundles form the strand space execution model. Bundles $\mathcal{B}$ have "forgotten" the metric that governs events in spacetime, and retained only the strand structure and communication arcs.

Each bundle $\mathcal{B}$ has a partial order $\preceq_{\mathcal{B}} = (\rightarrow \cup \Rightarrow)^*$, which is the weakest reflexive, transitive relation that extends the succession relation $\Rightarrow$ of nodes on the same strand, and extends the communication relation $\circ \stackrel{m}{\rightarrow} \circ$.

The acyclicity justifies a well-founded induction principle on $\preceq_{\mathcal{B}}$: If S is a non-empty set of nodes of $\mathcal{B}$, then there exist nodes in S that are $\preceq_{\mathcal{B}}$-minimal in S. Reasoning in strand spaces is ultimately justified by taking cases on these minimal nodes, given the permissible regular strands and adversary strands.

Hence, strand spaces are particularly natural for reasoning about distance-bounding protocols. The pure protocol analysis allows us to characterize the bundles a protocol allows. These then may be embedded in spacetime in any way that respects their causal structure, including the physical principle that causality cannot propagate faster than the speed of light. If this implies that the distance between two entities must have been below a selected bound d, then the protocol has achieved its goal.

Our overall strategy is akin to Meadows et al.'s 2007 work [24]. They capitalize on the causal characteristics of the challenge-response principles that govern security protocol correctness in general, which thereby determine how events can be ordered given the effects some of them must have on others. We add a particular realization of these principles for a model of security protocols [14]. Since that model is backed by an efficient tool, namely CPSA, we can apply the method on an industrial scale.

Mauw et al. [22] also observe the value of using the causal structure to guide protocol analysis for bounding distance. A separate source brought the problem back to our attention: Andre Scedrov and Carolyn Talcott discussed their work on distance-bounding protocols, including round-off attacks, repeatedly at the Protocol Exchange we periodically share [2,3,17,18]. The opportunity for an analysis of the kind we will present here was a consequence of those discussions, together with some preliminary work [33].

CPSA, a Cryptographic Protocol Shapes Analyzer. The protocol analysis tool CPSA implements the *enrich-by-need* method [14,28,29]. CPSA carries out protocol analysis by showing the analyst all of the minimal, essentially different executions compatible with some scenario of interest, often a very small set. By a scenario, we generally mean a situation in which some protocol roles have executed at least part way, with some assumptions that some parameters are freshly chosen, or some long-term keys are uncompromised. A *skeleton* means a formal representation of such a scenario.

Starting from a skeleton $\mathbb{A}_0$, CPSA systematically explores how to add new role instances and other information in ways that would help explain executions. CPSA does not explicitly represent adversary actions, but simply keeps track of what the adversary can obtain from the regular transmissions, subject to the assumptions. Mathematically, CPSA explores skeletons by rising in a *homomorphism ordering*, and it stops along any branch of its exploration when it runs out of possible explanations or reaches a *realized* skeleton.

A skeleton $\mathbb{B}$ is *realized* if, together with adversary actions compatible with its freshness and non-compromise assumptions, it can form a bundle $\mathcal{B}$. We say that $\mathbb{B}$ is *a skeleton of* such a bundle $\mathcal{B}$, and we say that a skeleton $\mathbb{A}$ *covers* bundle $\mathcal{B}$ if there exists a realized skeleton $\mathbb{B}$ such that $\mathbb{B}$ is a skeleton of $\mathcal{B}$, and a homomorphism $H\colon \mathbb{A} \to \mathbb{B}$.

The set of minimal realized skeletons are called the *shapes* for the starting skeleton $\mathbb{A}_0$. CPSA is useful because well-designed protocols often lead to small sets of shapes, even though the set of shapes is large or infinite in unfavorable cases.[1] CPSA presents the shapes in a concrete, graphical form, allowing a logically naive designer to understand the effects of varying protocol choices.

Moreover, each shape contains the events and their ordering needed for the non-metric, causal aspects of our distance-bounding analyses.

CPSA now allows assuming that certain messages pass over channels that ensure confidentiality or integrity. Any protocol implementation must discharge

[1] Indeed, since Andre et al. [13] proved the underlying problem class to be undecidable, uniform termination is impossible.

the assumptions, for instance by suitable cryptography. But CPSA can infer the effects of the assumptions, independent of particular choices about how to discharge them. We will use these channel assumptions in Section 4.

Protocol goals as formulas. CPSA offers a logical language $\mathcal{L}_{\Pi}$ to express goals for a protocol Π [15,27]. $\mathcal{L}_{\Pi}$ includes the following types of predicates:

- For each role $\rho \in \Pi$, and for each transmission or reception position i along ρ, $\mathcal{L}_{\Pi}$ contains a one-place predicate $r_{\rho,i}(n)$ that asserts that a node n is an instance of the i^{th} event along role ρ.
- For each role $\rho \in \Pi$, and for each parameter or variable x that helps to determine ρ's instances, $\mathcal{L}_{\Pi}$ contains a two-place predicate $p_{\rho,x}(n, v)$ that asserts that node n's instance for the x parameter is v.
- The causal ordering $n \prec n'$ is expressed by a predicate $\mathtt{prec}(n, n')$.
- Two nodes on the same strand satisfy the *collinear* predicate $\mathtt{coll}(n, n')$.
- $\mathtt{unique}(v)$ is satisfied if v is fresh; $\mathtt{non}(k)$, if key k is non-compromised.
- Confidentiality and integrity for a channel c are $\mathtt{conf}(c)$ and $\mathtt{auth}(c)$.

Any skeleton $\mathbb{A}_0$ may be expressed by a conjunctive formula of $\mathcal{L}_{\Pi}$. Thus, a CPSA run starting from $\mathbb{A}_0$ determines what must be true in all Π-bundles satisfying this formula, which we call the *characteristic formula* $\mathsf{cf}(\mathbb{A}_0)$ of $\mathbb{A}_0$.

A *goal formula* is a universally quantified implication $\forall \overline{x} \,.\, \Phi \Longrightarrow \bigvee_{i \in I} \exists \overline{y}_i \,.\, \Psi_i$, where Φ and the Ψ_i are conjunctions of atomic formulas (see Def. 4).

The special case $I = \emptyset$ gives the empty disjunction $\bigvee_{i \in \emptyset}$ with no way to be true, i.e. false. A goal $\mathsf{cf}(\mathbb{A}_0) \Longrightarrow \mathsf{false}$ states that no Π-bundle exhibits the scenario $\mathbb{A}_0$. If $\mathbb{A}_0$ assumes some putative secret k is heard unprotected, expressed in a parameter predicate $p_{\mathtt{lsn},x}(n, k)$ for a special role, the conclusion false ensures non-disclosure. Formulas with non-empty conclusions express authentication properties. They say that the behavior in the hypothesis Φ requires additional behavior found in one of the conclusions Ψ_i.

Indeed, a terminating run of CPSA may be summarized as a formula, which we call a *shape analysis formula* [27]. Suppose, starting from the initial scenario $\mathbb{A}_0$, CPSA terminates with the family of shapes $\{\mathbb{B}_i\}_{i \in I}$. It has discovered the security goal formula $\mathsf{cf}(\mathbb{A}_0) \Longrightarrow \bigvee_{i \in I} \exists \overline{y}_i \,.\, \mathsf{cf}(\mathbb{B}_i)$; the homomorphisms from $\mathbb{A}_0$ to the $\mathbb{B}_i$ determine the quantified variables $\overline{y}_i$. The formula must be true because the CPSA search is *sound*, i.e. it refines any skeleton $\mathbb{A}$ to a set of skeletons that cover all of the executions that $\mathbb{A}$ covers. Moreover, it is a strongest goal with the hypothesis $\mathsf{cf}(\mathbb{A}_0)$, because each of the shapes $\mathbb{B}_i$ really is an essentially different scenario that can occur. No correct goal could rule any of them out.

Thus, the shape analysis formula is the *strongest* security goal achieved by Π for this hypothesis [15,32]. In this way, CPSA allows us to discover what security goals Π achieves, for the situations of concern to us.

As this suggests, there is a natural ordering on security goals that share the same antecedent Φ, namely the entailment ordering on their conclusions $\bigvee_{i \in I} \exists \overline{y}_i \,.\, \Psi_i$. There is also a dual ordering on security goals that share the same *conclusion* $\overline{\Psi}$. Namely, of two security goals $\Gamma_1 = \Phi_1 \Longrightarrow \overline{\Psi}$ and $\Gamma_2 = \Phi_2 \Longrightarrow \overline{\Psi}$, Γ_1 is at least as strong as Γ_2 iff Φ_2 entails Φ_1.

In Section 4 we use this idea to compare different protocols, according to whether their shared distance-bounding conclusions require stronger or weaker assumptions to assure. The cross-protocol use of formulas like this is justified in our work on protocol transformation [15,32].

2 Adapting the strand model for distance-bounding

Let d be the usual Euclidean distance and c be the speed of light. We add metric information to bundles in the simplest way:

Definition 1. *Let $\mathcal{B}$ be a bundle, and let $E\colon \mathsf{nodes}(\mathcal{B}) \to \mathbb{R}^4$ be a function from the nodes of $\mathcal{B}$ into spacetime. $(\mathcal{B}, E)$ is a* spacetime bundle *iff, for all n_1, n_2 such that $n_1 \prec_{\mathcal{B}} n_2$, letting $E(n_1) = (t_1, x_1, y_1, z_1)$, and $E(n_2) = (t_2, x_2, y_2, z_2)$:*

1. *$t_1 < t_2$; and*
2. *$d((x_2, y_2, z_2), (x_1, y_1, z_1)) < c \cdot (t_2 - t_1)$.*

We will write $d_E(n_2, n_1)$ for $d((x_2, y_2, z_2), (x_1, y_1, z_1))$ and $t_E(n_1)$ for t_1.

Evidently, every spacetime bundle determines a (non-metric) bundle, namely its first component. Indeed, intuitively, however the events of $\mathcal{B}$ have occurred in space and time, they will satisfy conditions 1–2.

Conversely, any bundle $\mathcal{B}$ may be embedded into spacetime, i.e. it is the first component of some spacetime bundle:

Lemma 1. *Let $\mathcal{B}$ be a bundle. There exists an $E\colon \mathsf{nodes}(\mathcal{B}) \to \mathbb{R}^4$ such that $(\mathcal{B}, E)$ is a spacetime bundle.*

Proof. We choose to let each strand be stationary. Construct E by well-founded recursion on $\preceq_{\mathcal{B}}$. For each n choose a time $t_E(n)$ that exceeds the time of its immediate predecessors enough to allow its incoming messages to arrive. There is no upper bound on the choice for $t_E(n)$. □

Any skeleton $\mathbb{A}$ is compatible with or *covers* a (possibly empty) set of spacetime bundles $(\mathcal{B}, E)$, namely all those where there is a $H\colon \mathbb{A} \rightarrowtail \mathbb{B}$ such that $\mathbb{B}$ is a skeleton of $\mathcal{B}$.

Definition 2. *Let $\mathbb{A}$ be a skeleton with collinear nodes $n_1 \Rightarrow^+ n_2$, and let n' be a node. We say n_1, n_2* bound separation *from n' in $\mathbb{A}$ iff $n_1 \preceq_{\mathbb{A}} n' \preceq_{\mathbb{A}} n_2$.*

Lemma 2. *Let $(\mathcal{B}, E)$ be a spacetime bundle; $H\colon \mathbb{A} \rightarrowtail \mathbb{B}$, and $\mathbb{B}$ be a skeleton of $\mathcal{B}$. If n_1, n_2 bound separation from n' in $\mathbb{A}$, then*

$$d_E(H(n_1), H(n')) + d_E(H(n'), H(n_2)) < c \cdot (t_E(H(n_2)) - t_E(H(n_1))).$$

That is, using a local clock along the strand of $H(n_1)$, the principal executing it can bound the distance to the node $H(n')$. Thus, reasoning about ordering in a skeleton gives a uniform way to bound the distance between corresponding events in all the spacetime bundles it covers.

Proof. The homomorphism H preserves the ordering relations, as does the embedding of the realized skeleton $\mathbb{B}$ into the bundle $\mathcal{B}$. Thus, condition 2 in Def. 1 yields the desired inequality. □

We express requirements on distance-bounding protocols as security goals. Since we must talk about particular formulas and free variables, we will write formal variables $\lceil n \rceil$ with a ceiling in the next few paragraphs to distinguish them from our informal variables n ranging over nodes. Subsequently, we will revert to the usual ambiguity between mentioning formal variables and using informal variables. To express bounded separation goals, we distinguish particular formal variables $\lceil n_1, n_2, n' \rceil$.

Definition 3. *Let Γ be a security goal $\forall \overline{x}\,.\, \Phi \Longrightarrow \bigvee_{\ell \in L} \exists \overline{y}_\ell\,.\, \Psi_\ell$ in $\mathcal{L}_\Pi$ with non-empty L; let $\lceil n_1, n_2 \rceil$ be node variables among the variables $\overline{x}$, and $\lceil n' \rceil$ be among the variables $\overline{y}_\ell$ for* every *ℓ. Then:*

1. *$\Gamma, \lceil n_1, n_2, n' \rceil$ is a* distance-bounding requirement *for protocol Π (or a* requirement*, for short).*
2. *Π* achieves *the requirement $\Gamma, \lceil n_1, n_2, n' \rceil$, iff, for every realized Π-skeleton $\mathbb{B}$ and each variable assignment η such that η satisfies $\mathbb{B} \models_\eta \Phi$, there is some $\ell \in L$ and an η' extending η such that $\mathbb{B} \models_{\eta'} \Psi_\ell$ and moreover $\eta'(\lceil n_1 \rceil), \eta'(\lceil n_2 \rceil)$ bound separation from $\eta'(\lceil n' \rceil)$ in $\mathbb{B}$.*

Since both the conclusions Ψ_ℓ and bounding separation are preserved by homomorphisms, as soon as they are satisfied in a branch of a CPSA, they will remain true thereafter. Moreover, by Lemma 2, the requirement ensures that some strand satisfying Ψ_ℓ will be no farther from a strand satisfying Φ than the locally elapsed time $\Delta t \cdot c/2$ between the i^{th} and j^{th} node.

Hence, suppose we want to check if Π achieves a requirement $\Gamma, \lceil n_1, n_2, n' \rceil$, where Γ is of the form $\forall \overline{x}\,.\, \Phi \Longrightarrow \bigvee_{\ell \in L} \exists \overline{y}_\ell\,.\, \Psi_\ell$, and Φ is the characteristic formula of a CPSA starting scenario $\mathbb{A}_0$, i.e. $\Phi = \mathsf{cf}(\mathbb{A}_0)$.

1. Execute CPSA starting from the scenario $\mathbb{A}_0$, obtaining the set of shapes $\{H_\ell \colon \mathbb{A}_0 \to \mathbb{B}_\ell\}_{\ell \in I}$;
2. ascertain that each $\mathbb{B}_\ell \models_\eta \Gamma$;
3. for the satisfying variable assignments η', check that $\eta'(\lceil n_1 \rceil), \eta'(\lceil n_2 \rceil)$ bound separation from $\eta'(\lceil n' \rceil)$ in $\mathbb{B}_\ell$.

In the favorable case in which I is finite, these steps terminate.

We can easily express bounded separation as a conjunctive formula in the variables $\lceil n_1, n_2, n' \rceil$, namely:

$$\mathtt{prec}(\lceil n_1 \rceil, \lceil n' \rceil) \wedge \mathtt{prec}(\lceil n' \rceil, \lceil n_2 \rceil),$$

which we will denote $\mathtt{bnd\_sep}(\lceil n_1 \rceil, \lceil n_2 \rceil, \lceil n' \rceil)$. Thus, in practice we perform surgery on the given goal Γ to obtain Γ^+:

$$\forall \overline{x}\,.\, \Phi \Longrightarrow \bigvee_{\ell \in L} \exists \overline{y}_\ell\,.\, (\Psi_\ell \wedge \mathtt{bnd\_sep}(\lceil n_1 \rceil, \lceil n_2 \rceil, \lceil n' \rceil)).$$

CPSA can check this security goal directly, as we illustrate in the next section.

3 Examples

In this section, we show how CPSA is used to find and fix a flaw in the Terrorist-fraud Resistant and Extractor-free Anonymous Distance-bounding (TREAD) protocol [4]. The aim of its authors is to "obtain provable terrorist-fraud resistant protocols without assuming that provers have any long-term secret key". Alas, the case in which TREAD is implemented with public key cryptography as shown in Fig. 2 of [4] has an authentication failure.

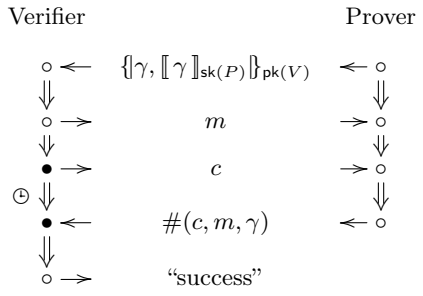

Fig. 1. TREAD Protocol

Fig. 1 shows our model of the TREAD protocol. Each participant, V and P, has a public key $\mathsf{pk}(\cdot)$ and a private key $\mathsf{sk}(\cdot)$. A message is encrypted with $\{\!|\cdot|\!\}_{\mathsf{pk}(\cdot)}$ and signed with $[\![\cdot]\!]_{\mathsf{sk}(\cdot)}$. The first message exchanged in the protocol is γ signed by the prover and then encrypted for the verifier.

All distance-bounding protocols include a *fast phase*, where one principal measures the time it takes for a sequence of message interactions. Our modeling of TREAD abstracts away details of its fast phase by a pair of messages. The two bullets • near the clock ⏲ in the Verifier role show the beginning and end of the timed fast phase.

In TREAD, γ is a pair of random n-bit values $\gamma = (\alpha, \beta)$. During the fast phase of the protocol, the Verifier sends n one bit messages that make up the contents of randomly chosen n-bit message c. The Prover responds to the reception of c_i with r_i, where

$$r_i = \begin{cases} \alpha_i & \text{if } c_i = 0, \\ \beta_i \oplus m_i & \text{if } c_i = 1. \end{cases}$$

The Verifier declares success if it receives the responses it expects within the protocol's time bound. If the adversary cannot obtain γ, the adversary is highly unlikely to provide the right n values for the r_i. In that case, n bounded separation claims are likely to hold.

In our protocol representation with a single fast exchange, the Prover sends the hash of c, m, and γ, and the Verifier declares success if it receives that message. Thus, in our version, we would like bounded separation to hold where n_1, n_2 are the two Verifier nodes on the timed edge, and n' is the Prover node that transmits $\#(c, m, \gamma)$. The security goal Γ asserts that if a Verifier run completes, a Prover run with matching V, P, γ, m, c parameters should also complete.

Analysis of TREAD Fig. 1 describes the TREAD protocol. Consider the point-of-view in which the Verifier has run to completion with freshly chosen m, c and non-compromised $\mathsf{sk}(P), \mathsf{sk}(V)$. What else must have happened?

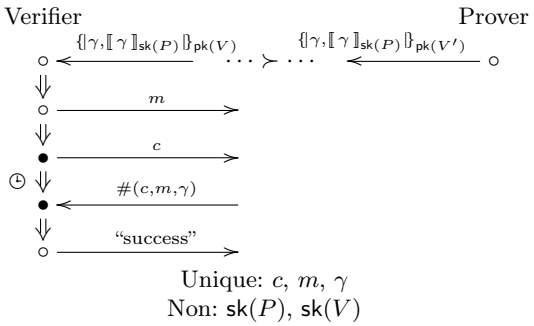

Fig. 2. TREAD Shape

The shape found by CPSA is displayed in Fig 2. CPSA infers that the Prover was active, but it may only have transmitted its first message, which may have been altered before delivery by the adversary. The message received at the Verifier's 4^{th} node can be synthesized by the adversary. CPSA is telling us that there are bundles that are compatible with the shape in which adversary strands synthesize all the messages received by the Verifier using only the message sent at the Prover's first (and only) node. Thus, neither Γ nor the bounded separation property holds.

CPSA explains each step it takes on its way to finding its answers, and a knowledgeable user can use this information to fix the protocol. However, we press on, trying to fix the problem by adding a confirming message at the end of the protocol. This is a reasonable thing to try, as at least one industrial protocol uses this technique to (slowly) authenticate the replies sent during the fast phase [8]. Plus, it is intuitively clear that this should allow the Verifier to conclude that the Prover must have engaged in a fast phase.

Fig. 3 shows the amended protocol we call TREAD+. When started with the point-of-view skeleton in which the Verifier runs to completion, CPSA finds the shape in Fig. 3. This time, CPSA concludes that the Prover must have run to full length. However, the mismatch between the first message sent by the Prover and received by the Verifier is still present, so Γ fails. Message $\#(c, m, \gamma)$ is received by the Verifier on the second timed node, and is also sent by the Prover. However, CPSA does not report that the transmission has to precede the reception: The adversary can synthesize $\#(c, m, \gamma)$ before the Prover sends it! This occurs because the Prover's random values leak in the first message. Thus, bounded separation again fails.

The RETREAD protocol fixes the authentication problem in the TREAD protocol. It alters the first message by including the name of the Verifier, V, in the signed part of the message. Therefore, the first message in both roles of the protocol is $\{|\gamma, [\![\gamma, V]\!]_{\mathsf{sk}(P)} |\}_{\mathsf{pk}(V)}$. When CPSA is started with the point-of-view skeleton in which the Verifier runs to completion, it finds the shape in Fig. 4.

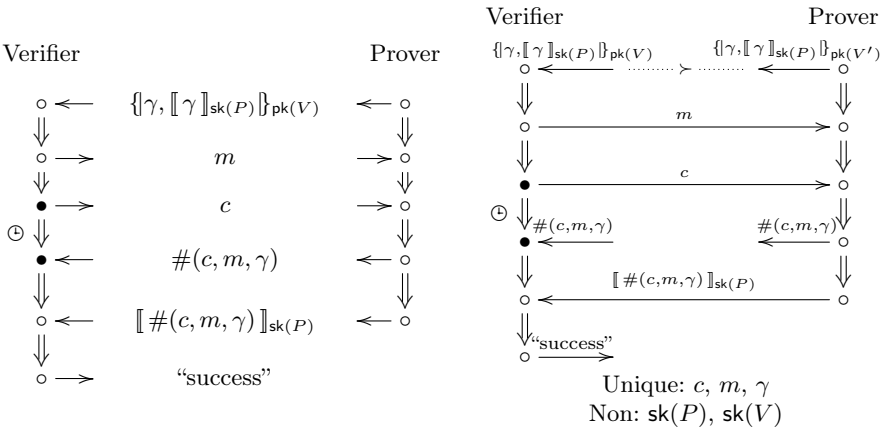

Fig. 3. TREAD+ Protocol (l) and relevant shape (r)

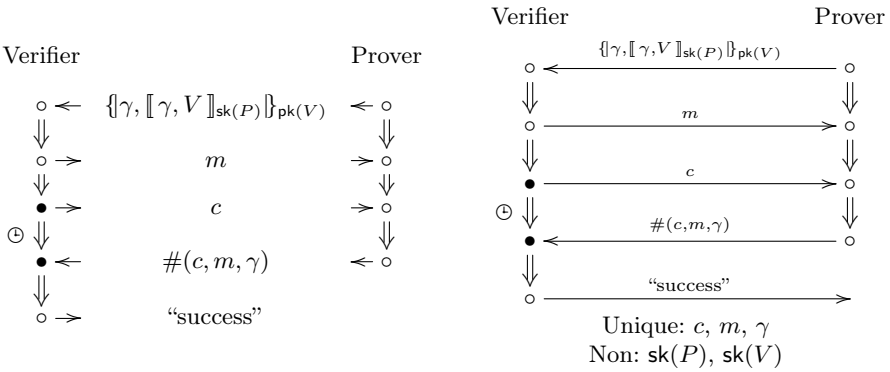

Fig. 4. The RETREAD protocol (l) and its shape (r)

No adversary behavior need occur in bundles compatible with this shape. What CPSA learns is expressed in the shape analysis sentence:

If a Verifier with parameters P, V, c, m, and γ runs to completion, and
- c, m, and γ are assumed to be uniquely originating, and
- $\mathsf{sk}(P)$ and $\mathsf{sk}(V)$ are assumed to be non-originating,

then a Prover with parameters P, V, c, m, and γ ran to completion, with bounded separation for the timed Verifier nodes and the 3$^{\text{th}}$ Prover node.

Adding V's name inside the signature in the Prover's last message in TREAD+ also forces V and P to agree on V's identity, ensuring γ remains secret and ensuring bounded separation. However, RETREAD is a superior protocol since it is shorter and requires only a single signature.

4 Taxonomy

Much of the recent literature on symbolic analysis of distance-bounding protocols has focused on classifying protocols according to their ability to resist various kinds of attacks (e.g. [7,9,22]). Our position is that it is more useful to categorize protocols according the security goals they achieve. We follow the approach from [32] in which security goals are expressed as first-order logical formulas. The strength ordering of goal formulas is naturally captured by logical implication. If Γ_1 and Γ_2 are security goals, then Γ_1 is at least as strong as Γ_2 iff $\Gamma_1 \Rightarrow \Gamma_2$.

Definition 4. *A* security goal *is a closed formula $\Gamma \in \mathcal{L}_\Pi$ of the form*

$$\forall \bar{x} \,.\, (\Phi \Longrightarrow \bigvee_{k \in K} \exists \bar{y}_k \,.\, \Psi_k)$$

where Φ and Ψ are conjunctions of atomic formulas. We write $\mathsf{hyp}(\Gamma) = \Phi$ *and* $\mathsf{conc}(\Gamma) = \bigvee_{k \in K} \exists \bar{y}_k \,.\, \Psi_k$.

Fundamentally, all distance-bounding protocols have the same minimal goal. If the verifier accepts a run apparently with prover P, then P must have responded to the challenge after the start of the fast phase of the protocol and before its completion. Protocols may have more stringent authentication requirements such as needing the prover to agree on the verifier's name and other authenticated data. But often such agreement is achieved in the service of the main goal which is to ensure P must be close. We can naturally express this in our goal language.

To say that the verifier has accepted a run apparently with P we may write

$$\Phi_1(n, P) = \texttt{VerifierDone}(n) \wedge \texttt{Prover}(n, P)$$

where $\texttt{VerifierDone}(\cdot)$ is a predicate that holds for the last node of a verifier's run, and $\texttt{Prover}(\cdot)$ signifies the verifier's value for the prover's identity.

To express the requirement that P respond to the verifier's challenge during the fast phase, we need to identify the nodes starting and stopping the fast phase. We can write

$$\Phi_2(n_1, n_2) = \texttt{StartTimer}(n_1) \wedge \texttt{StopTimer}(n_2) \wedge \texttt{coll}(n_1, n_2)$$

where $\texttt{StartTimer}(\cdot)$ and $\texttt{StopTimer}(\cdot)$ serve to identify the nodes starting and stopping the fast phase on the verifier's strand. $\texttt{coll}(n_1, n_2)$ states that these nodes start and stop the fast phase on the same strand. We similarly must ensure that we are referring to the fast phase of the same strand as the one accepting the run with P. It suffices to express that n and n_1 are on the same strand. Putting it all together, we have:

$$\Phi(n, P, n_1, n_2) = \Phi_1(n, P) \wedge \Phi_2(n_1, n_2) \wedge \texttt{coll}(n, n_1) \tag{1}$$

Equation 1 serves as the hypothesis for the distance-bounding security goal. The conclusion must state that P responded to the challenge during the fast

phase. As all distance-bounding protocols have an event in which the prover sends the reply to a challenge, we use $\texttt{ProverReply}(\cdot)$ to denote such a node of a prover strand. We again use $\texttt{Prover}(\cdot,\cdot)$ to express that the prover's identity for the $\texttt{ProverReply}$ node is P. Finally, we use $\texttt{bnd\_sep}$ to express the ordering required for the fast phase. The result is:

$$\Psi(n_1, n_2, n', P) = \texttt{ProverReply}(n') \wedge \texttt{Prover}(n', P) \wedge \texttt{bnd\_sep}(n_1, n_2, n') \quad (2)$$

The basic distance-bounding security goal is thus:

$$\mathsf{DB} = \forall n, P, n_1, n_2 \,.\, \Phi(n, P, n_1, n_2) \Longrightarrow \exists n' \,.\, \Psi(n_1, n_2, n', P) \quad (3)$$

However, no protocol can achieve DB as formulated in Eq. 3. The assumptions in $\mathsf{hyp}(\mathsf{DB})$ are too weak to imply bounded separation. First, distance bounding is hopeless unless the verifier chooses fresh values. We will henceforth always assume this, adopting a corresponding strengthening Φ' in place of Eq. 1.

But also, Φ makes no assumption about the authenticity or confidentiality of any communications channels—either directly or through assumptions on cryptographic keys. It is well-known that authentic or confidential channels cannot be constructed without access to an authentic or confidential channel [21], or corresponding secret keys. We identify a collection of additional assumptions that can help to ensure a protocol can achieve the goal of bounding the distance of the apparent prover. We identify three main types of assumptions:

- s. Secrecy of long-term keys (private keys and/or shared symmetric keys)
- f. Freshness of prover-chosen values
- a. Authenticity of messages received during the fast phase

The assumption that a given long-term key has been kept secret is familiar for cryptographic protocols of all types. In surveying the literature, there are typically three types of long-term keys that distance-bounding protocols tend to rely on: private keys belonging to the prover ($\mathsf{sk}(P)$), private keys belonging to the verifier ($\mathsf{sk}(V)$), and symmetric keys shared by the prover and verifier ($\mathsf{ltk}(P, V)$). We may state the corresponding secrecy requirements as follows:

$$\mathsf{s}_{\mathsf{ltk}}(n, P, V) = \texttt{Verifier}(n, V) \wedge \texttt{Prover}(n, P) \wedge \texttt{non}(\mathsf{ltk}(P, V)) \quad (4)$$

$$\mathsf{s}_{\mathsf{prv}}(n, P) = \texttt{Prover}(n, P) \wedge \texttt{non}(\mathsf{sk}(P)) \quad (5)$$

$$\mathsf{s}_{\mathsf{vrf}}(n, V) = \texttt{Verifier}(n, V) \wedge \texttt{non}(\mathsf{sk}(V)) \quad (6)$$

The freshness of prover-chosen values can also play an important role in the success of distance-bounding protocols. If a verifier believes the prover's nonces to be randomly chosen and shared only with the verifier, then by incorporating the nonces into the reply during the fast phase the verifier can conclude it really is the prover providing the reply. However, there are several natural reasons this assumption may not be justified. In many distance-bounding protocols the prover has very limited computational power, and so may also not have a reliable source of randomness. It may also be the case that a dishonest and distant prover

is willing to share their nonces with a malicious accomplice who is in physical proximity with the verifier. This is related to Terrorist Fraud Attacks [11], about which we say more in a later section. We may state the freshness assumption on a prover's nonce as follows:

$$\mathsf{f}(n, np) = \texttt{ProverNonce}(n, np) \wedge \texttt{uniq}(np) \tag{7}$$

In some protocols (e.g. TREAD), the prover contributes two nonces. In those cases, for each of the nonces, f will include a pair of conjuncts like Eq. 7. In our analysis below, we always make the same assumption on all of the prover's nonces. That is, we either assume all nonces are fresh, or we don't assume any are.

Finally, some protocols might be run in environments where it is reasonable to assume that only regular (i.e. honest) provers can provide the replies during the fast phase. Consider, for example, a secure facility that enforces physical access control to a building that uses a distance-bounding protocol to gate access to special rooms. The fast phase may use near-field communication meaning that only those provers who have already passed the initial access control would be within range. This is one way to ensure the authenticity of messages received during the fast phase.

Whether malicious parties have access to the timed channel inbound to the verifier is related to Distance Hijacking Attacks [9]. We may state the assumption that the inbound timed channel is authentic as follows:

$$\mathsf{a}(n, \mathit{timed}) = \texttt{TimedChannel}(n, \mathit{timed}) \wedge \texttt{auth}(\mathit{timed}) \tag{8}$$

Equations 4-8 allow us to define a family of distance-bounding security goals according to which subsets of the assumptions we include. Many protocols only require the prover to have access to a single long-term key. Depending on whether it is a shared symmetric key or a private signing key, we would use either Eqn. 4 or 5. By making all possible combinations of assumptions of type s, f, and a, we naturally generate eight possible goals which we denote $\mathsf{DB}^{\mathcal{P}(\{\mathsf{sfa}\})}$ according to which subset of $\{\mathsf{s}, \mathsf{f}, \mathsf{a}\}$ is included in the assumptions of $\mathsf{hyp}(\mathsf{DB}^{\mathcal{P}(\{\mathsf{sfa}\})})$ So, for example, $\mathsf{DB} = \mathsf{DB}^{\emptyset}$ because we make none of the assumptions. The goal that only assumes authenticity of messages received during the fast phase is denoted DB^{a}, (with the set braces suppressed for readability) which stands for the formula:

$$\forall n, P, n_1, n_2, \mathit{timed} \;.\; \Phi(n, P, n_1, n_2) \wedge \mathsf{a}(n, \mathit{timed}) \Longrightarrow \exists n' \;.\; \Psi(n_1, n_2, n', P).$$

Following the ideas in [32], this family of goals is naturally ordered by implication. Goal formulas that make fewer assumptions are naturally stronger. Figure 5 depicts the ordering of the family $\mathsf{DB}^{\mathcal{P}(\{\mathsf{sfa}\})}$. Only the superscripts are denoted in the diagram. This partial ordering can serve as a yard stick to measure the relative strength of a variety of designs for distance-bounding protocols. If a protocol satisfies the goal at one point in the partial order, then it satisfies all goals below it (since they are ordered by implication). Therefore, we can evaluate protocols

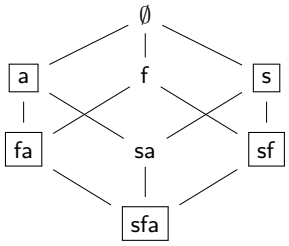

Fig. 5. Strength ordering for $\mathsf{DB}^{\mathcal{P}(\{\mathsf{sfa}\})}$. Boxes indicate maximal strength achieved by at least one protocol from our survey.

according to the maximal goals they achieve. We performed a survey of numerous protocols from the literature, and the boxes indicate the maximal strength achieved by at least one of the protocols we studied.

For protocols such as TREAD that rely on two long-term keys, one of the keys is typically the prover's signing key, while the other is either the verifier's private decryption key ($\mathsf{sk}(V)$) or a shared symmetric key ($\mathsf{ltk}(P,V)$). We may wish to separate the assumptions we make about their secrecy. This yields a bigger family of goals denoted $\mathsf{DB}^{\mathsf{ss'fa}}$ where s represents the assumption $\mathsf{s}_{\mathsf{prv}}$ (Eqn. 5), and s' represents either $\mathsf{s}_{\mathsf{ltk}}$ (Eqn. 4) or $\mathsf{s}_{\mathsf{vrf}}$ (Eqn. 6) depending on the design of the protocol. This yields a bigger lattice of security goals depicted in Fig. 6. Again, the boxes indicate maximal strengths achieved by at least one protocol among those we surveyed.

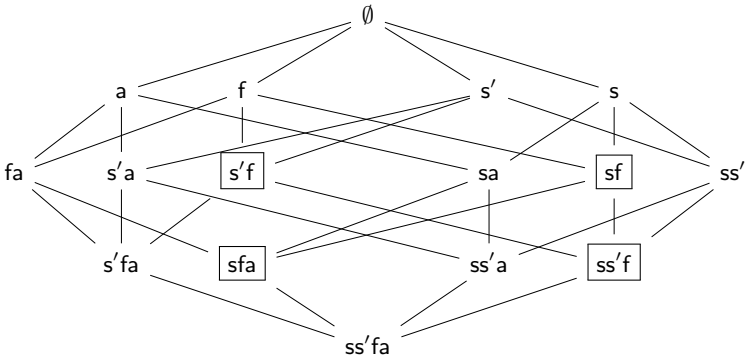

Fig. 6. Strength ordering for $\mathsf{DB}^{\mathcal{P}(\{\mathsf{ss'fa}\})}$. Boxes indicate maximal strength achieved by at least one protocol from our survey.

As stated above, we analyzed numerous protocols from the literature. Our intent is not to be exhaustive, but rather to demonstrate the utility of assumption-based analyses for comparing the relative strength of different designs of distance-bounding protocols. Space constraints preclude an exhaustive description of all the analyses, but the results are summarized in Table 1[2] and we discuss a few noteworthy highlights below. The times reported are based on runs using a 2018

[2] Cf. `https://github.com/mitre/cpsaexp/tree/master/doc/dist_bnd_prots`.

MacBook Air with 1.6GHz Dual-Core Intel Core i5 processor with 16GB of RAM. They represent the total elapsed time after verifying all 8 or 16 variants of the goal depending on how many long-term keys the protocol uses. As the table makes clear, CPSA is an extremely efficient tool.

Table 1. Various distance-bounding protocols ordered by strength.

Protocol	**Strength**	**Elapsed Time (s)**
Protocols with a single long-term key		
Hancke and Kuhn [16]	$\{\mathsf{s}\}, \{\mathsf{a}\}$	0.03
Kim and Avoine [19]	$\{\mathsf{s}\}, \{\mathsf{a}\}$	0.03
Munilla et al. [26]	$\{\mathsf{s}\}, \{\mathsf{a}\}$	0.07
Reid et al. [31]	$\{\mathsf{s}\}, \{\mathsf{a}\}$	0.04
Swiss-Knife [20]	$\{\mathsf{s}\}, \{\mathsf{a}\}$	0.05
Mauw et al. [23]	$\{\mathsf{sf}\}, \{\mathsf{a}\}$	0.03
Meadows et al. [24]	$\{\mathsf{sf}\}, \{\mathsf{fa}\}$	0.05
BC-Signature [6]	$\{\mathsf{sf}\}$	0.06
CRCS [30]	$\{\mathsf{sf}\}$	0.06
BC-FiatShamir [6]	$\{\mathsf{sfa}\}$	0.10
Protocols with two long-term keys		
Paysafe [8]	$\{\mathsf{sf}\}, \{\mathsf{s'f}\}$	0.12
TREAD-SK [5]	$\{\mathsf{sfa}\}, \{\mathsf{s'f}\}$	0.08
TREAD-PK [5]	$\{\mathsf{sfa}\}$	0.05
TREAD variants introduced in this paper		
TREAD-SK+	$\{\mathsf{sfa}\}, \{\mathsf{s'f}\}$	0.16
RETREAD-SK	$\{\mathsf{sfa}\}, \{\mathsf{s'f}\}$	0.07
RETREAD-PK	$\{\mathsf{sfa}\}, \{\mathsf{ss'f}\}$	0.06
TREAD-PK+	$\{\mathsf{sfa}\}$	0.16

We first note that it seems to be easy for distance-bounding protocols to satisfy the weakest goal ($\mathsf{DB}^{\mathsf{sfa}}$ or $\mathsf{DB}^{\mathsf{ss'fa}}$). Every protocol we checked was secure under the strongest set of assumptions. The weakest protocol we discovered was Brands and Chaum's early adaptation of the Fiat-Shamir identification scheme into a distance-bounding scheme [6]. This weak result may not be entirely accurate, but might rather be an artifact of modeling algebraic properties with logical axioms.

At the other end of the spectrum, there is a collection of protocols that all satisfy both DB^{a} and DB^{s} which are incomparable goals [16,19,20,26,31]. Indeed, this is the best we can hope for. As we have already seen, $\mathsf{DB}^{\emptyset}$ is impossible to achieve due to the need to have access to at least one confidential or authentic channel [21]. DB^{f} is unsatisfiable for the same reason. Therefore, simultaneously satisfying both "shoulders" of Fig. 5 is the maximum strength possible.

It is instructive to consider the design principles used by various protocols that contribute to their strength or weakness. The family of protocols achieving the maximum strength are all based on the same core design. Namely, in

the fast phase, the prover's reply cryptographically binds the verifier's challenge with a long-term shared symmetric key that serves to authenticate the prover. The exact way in which these values are cryptographically bound varies widely, but it typically involves generating a hash of a message containing at least the shared key and the verifier's nonce. Much of the variation in the designs is attributable to the need to make the cryptographic operation as simple as possible. More computationally intensive operations force the verifier to accept longer threshold times for the round trip because the verifier needs to account for the computation time as well. Longer threshold times generally provide weaker distance guarantees. Our symbolic analysis only ensures that an upper bound on the distance can be achieved. Since it does not consider the computation time explicitly, CPSA does not distinguish among these protocols.

In order to better understand how different designs fall short of the maximum strength, consider the protocol from Mauw et al. [23]. Rather than creating an explicit binding between the long-term shared symmetric key and the verifier's nonce, they create an implicit binding. They do this with a message in the setup phase. During this first phase, the prover sends a nonce to the verifier encrypted under their long-term, shared symmetric key. During the fast phase, the prover combines the verifier's nonce with its own nonce from the first phase. This creates an implicit binding between the verifier's nonce and the long-term key. But, crucially, this binding only succeeds if the long-term key has not been compromised, *and* the prover's nonce is indeed random and fresh. An adversary near to the verifier who is capable of guessing the prover's nonce (or an adversary who can coerce the prover into leaking its nonce during the first phase) can cause a distant prover to appear close to the verifier. Thus, in this protocol, the security of the long-term key is not enough. The verifier must also assume the prover's nonce is not available outside the bounds of the protocol execution. This is why it does not achieve DB^{s}, but does achieve $\mathsf{DB}^{\mathsf{sf}}$. When considering the goal DB^{a}, the authenticity of the fast channel guarantees the prover is honest. Since the reply also contains the prover's identity, this authentic channel successfully authenticates the prover's identity.

The remaining protocols suffer in similar ways. Generally speaking, they also perform implicit bindings between the verifier's nonce and the long-term key. In attempting to make the prover's response as fast as possible to compute, various techniques are chosen that suffer from subtle algebraic collisions. For example, the bindings are frequently created by performing an xor operation which is very efficient. But such values are not inherently integrity protected, so there is an opportunity for algebraic manipulation. This is contrast to a standard hash function which may be slower to compute, but which does not admit such algebraic manipulations.

We also analyzed TREAD together with a shared-key version also present in [5]. We distinguish them as TREAD-PK and TREAD-SK respectively. We can now compare them to the altered versions we introduced in Sec. 3. Our earlier analysis focused solely on the goal $\mathsf{DB}^{\mathsf{ss'f}}$. While TREAD-PK fails to satisfy that goal, it does satisfy $\mathsf{DB}^{\mathsf{sfa}}$. This says that, when the verifier assumes its timed

inbound channel is authentic, the verifier need not rely on the secrecy of its own private key to achieve the distance bound. TREAD-SK additionally satisfies $\mathsf{DB}^{\mathsf{s'f}}$ which implies the goal we investigated in Sec. 3 but is also incomparable with the single strongest goal achieved by TREAD-PK. Although TREAD-SK does not satisfy any goals stronger than the strongest one achieved by TREAD-PK, it does satisfy goals the public key version does not. In this sense, TREAD-SK is strictly stronger than TREAD-PK.

Notice that TREAD-PK+ provides no benefit beyond TREAD-PK, and similarly for the shared-key versions. RETREAD-PK, on the other hand, is slightly stronger than TREAD-PK, although not quite as strong even as TREAD-SK since it doesn't achieve $\mathsf{DB}^{\mathsf{s'f}}$. RETREAD-SK satisfies the same goals as TREAD-SK, so adds no value in a shared key context.

5 Related Work

Spacetime vs. causality. A key aspect of our approach to analyzing distance-bounding protocols is the lack of any explicit account of time or distance in the protocol models. We are not the first to make the observation that a causality-based analysis is informative enough to draw conclusions about time and distance. We follow the ideas taken by Mauw et al. in [22] in which they introduce a semantic model that explicitly accounts for time and distance. They then relate that model to the execution model underlying Tamarin [25] just as we relate spacetime bundles and realized skeletons. The underlying Dolev-Yao adversary model of Tamarin is sufficient to capture all relevant attacks.

This is in contrast to the work of Chothia et al. [7], which uses specific classes of processes to model dishonest provers, instead of relying solely on the underlying Dolev-Yao processes. Their approach is also causality-based: Rather than model time and distance quantitatively, they model "places;" processes in the same place are nearby. They adapt the pi-calculus operational semantics [1] so that processes that communicate during the fast phase have the same place.

Various other symbolic approaches account for time and distance more explicitly [9,17,10]. Our core insight arose from discussions with Andre Scedrov and Carolyn Talcott when they presented their work starting with [17] at our Protocol Exchange meeting. Our discussions suggested we could separate the analysis of the causal constraints from an analysis of the quantitative constraints of time and distance. We believed we could first reason causally and collect quantitative constraints along the way. The causal reasoning would justify deriving a tolerable strand-local delay from the desired quantitative distance bound. Lemmas 1–2 justify the procedure.

Attack-based vs. assumption-based. Since the very early days of studying distance-bounding protocols the focus has been on preventing various types of attacks. The attacks are commonly referred to by names such as Terrorist Fraud and Mafia Fraud. One frequently finds intuitive definitions of these attack types based on the relative locations of Dolev-Yao attackers, honest & dishonest

provers, and the verifier. The informality of these intuitions can make it quite difficult to interpret how they should be formally defined in any given model. Indeed, as observed in [22], there still remains some disagreement around the appropriate formal definition of Terrorist Fraud.

The clearest, most formal definitions we could find were in [7]. They introduce systematic definitions of dishonest provers for Mafia Fraud and Terrorist Fraud. They then explore the set of combinations of verifiers, Dolev-Yao attackers, and honest & dishonest provers in all possible relative locations, and organize them into a hierarchy of attacks.

In our view, focusing on and explicitly modeling different attack types runs counter to the spirit of most modern approaches to symbolic analysis of protocols. The community no longer makes distinctions about whether an adversary executes a reflection attack or a replay attack. The community no longer creates different protocol models for closed systems and open systems.

Our approach is based on a lesson learned by one of the authors from an observation made by Andre Scedrov regarding the classic Needham-Schroeder protocol. The standard view that Lowe found a previously undiscovered attack is somewhat misleading. The original protocol was secure *under the assumptions* made by Needham and Schroeder. Lowe's attack did not invalidate old security claims. It merely showed that the desired authentication property doesn't hold *under a weaker set of assumptions*. In the language of strand spaces, Needham and Schroeder assumed that initiators only engage in sessions with responders whose private keys are non-originating. Indeed, under such an assumption the protocol *does* achieve the desired conclusion. The question of whether such an assumption is justified is separate from that of whether the protocol achieves the right conclusion under the assumption.

This observation motivates the assumption-driven analysis in contrast to an attack-driven analysis. We believe our focus on altering assumptions instead of altering attacks helps focus attention on the security goals achieved by a distance-bounding protocol regardless of the type of attack. Of course, the assumptions one makes are closely related to the types of attacks considered. But we find the shift in perspective to be enlightening.

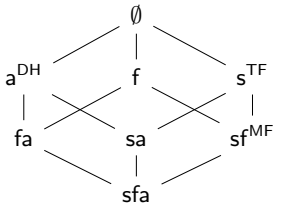

Fig. 7. Conjecture: attacks and the assumptions that prevent them.

Nevertheless, we have a conjecture connecting our lattice of assumptions to the standard attack types in the literature. Figure 7 annotates our assumption lattice for protocols using one long-term key with three attack types: Mafia Fraud (MF), Terrorist Fraud (TF), and Distance Hijacking (DH). For each attack type, we associate it with the weakest goal such that, if the protocol achieves that goal, then it resists the given attack type. So, for instance, if a protocol

achieves DB^{a} then it resists distance hijacking attacks. This association is only an informal conjecture at this point. The corresponding association for protocols with two long-term keys is less clear. It is worth noting, however, the the relative order of the attacks in Fig. 7 matches the corresponding order embedding in the attack hierachy of [7]. Establishing this conjecture, or making it more precise, would require a more careful comparison of the semantics of the formal models.

Symbolic vs. quantitative analysis. Our symbolic analysis relies on a simple and clear use of causal structures to infer the security goals achieved by distance-bounding protocols. However, the simplicity and clarity is often obtained by abstracting away the finer details of the fast phase. The fast phase typically involves repeated round trips of single-bit messages, which we represent as a single round trip of many-bit messages. The causal structure arises out of unique-origination assumptions on nonces which preclude any other agent from being able to send the nonce without first receiving it.

However, at the bit level, any given round trip does not guarantee the desired causal order because an adversary or a dishonest prover always has a chance of guessing the correct reply bit before receiving the challenge bit. The causal conclusions only emerge probabilistically over time as challenge-response round trips are performed. Symbolic analyses are therefore incapable of yielding insights about how many challenges a verifier should issue to be confident of the causal consequences. Recent work by Andre Scedrov and others explicitly addresses this question for the Hancke-Kuhn family of distance-bounding protocols [2,3].

Another creative line of inquiry by Scedrov and others [18] involves a more nuanced analysis of just how strongly the timing constraints can bound the distance between the verifier and the prover. They introduce a model that accounts not only for the time it takes for a message to travel through space, but also for the time it takes for instructions to execute. Because low-powered processors can often only perform one instruction during any given clock tick, there can be time between the event of starting the timer and the event of sending the challenge that is unaccounted for by the timing constraint. They discover the possibility of an "Attack Between the Ticks" in which a distant prover takes advantage of this time discrepency to appear much closer than they actually are.

6 Conclusion

In this paper we introduced a version of strand spaces that explicitly accounts for the physical properties of spacetime. We demonstrated that it is always possible to embed the standard strand space bundles into spacetime bundles in such a way that any quantitative constraints on distance and time are satisfied.

This justifies using CPSA without modifications to analyze the security of distance-bounding protocols, illustrated by analyzing and repairing the TREAD protocol which had previously been shown to be vulnerable to attack. A survey of various distance-bounding protocols from the literature places them in a taxonomy of protocols according to their strength. In contrast to the prevailing

trend, we organize our taxonomy not on the basis of attacks that are possible, but—dually—on the basis of the assumptions required for a verifier to bound the distance to a given prover.

We believe the shift in perspective to an assumption-based taxonomy from an attack-based one provides a clearer understanding of the conditions under which distance-bounding protocols succeed and fail.

References

1. Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *28th ACM Symposium on Principles of Programming Languages (POPL '01)*, pages 104–115, January 2001.
2. Musab A. AlTurki, Max I. Kanovich, Tajana Ban Kirigin, Vivek Nigam, Andre Scedrov, and Carolyn L. Talcott. Statistical model checking of distance fraud attacks on the hancke-kuhn family of protocols. In David Lie and Mohammad Mannan, editors, *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy, CPS-SPC@CCS 2018, Toronto, ON, Canada, October 19, 2018*, pages 60–71. ACM, 2018.
3. Musab A. AlTurki, Tajana Ban Kirigin, Max I. Kanovich, Vivek Nigam, Andre Scedrov, and Carolyn L. Talcott. A multiset rewriting model for specifying and verifying timing aspects of security protocols. In Joshua D. Guttman, Carl E. Landwehr, José Meseguer, and Dusko Pavlovic, editors, *Foundations of Security, Protocols, and Equational Reasoning - Essays Dedicated to Catherine A. Meadows*, volume 11565 of *Lecture Notes in Computer Science*, pages 192–213. Springer, 2019.
4. Gildas Avoine, Xavier Bultel, Sébastien Gambs, David Gérault, Pascal Lafourcade, Cristina Onete, and Jean-Marc Robert. A terrorist-fraud resistant and extractor-free anonymous distance-bounding protocol. In Ramesh Karri, Ozgur Sinanoglu, Ahmad-Reza Sadeghi, and Xun Yi, editors, *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 800–814. ACM, 2017.
5. Gildas Avoine, Xavier Bultel, Sébastien Gambs, David Gérault, Pascal Lafourcade, Cristina Onete, and Jean-Marc Robert. A terrorist-fraud resistant and extractor-free anonymous distance-bounding protocol. *IACR Cryptology ePrint Archive*, 2017:297, 2017.
6. Stefan Brands and David Chaum. Distance-bounding protocols (extended abstract). In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. Springer, 1993.
7. Tom Chothia, Joeri de Ruiter, and Ben Smyth. Modelling and analysis of a hierarchy of distance bounding attacks. In William Enck and Adrienne Porter Felt, editors, *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pages 1563–1580. USENIX Association, 2018.
8. Tom Chothia, Flavio D. Garcia, Joeri de Ruiter, Jordi van den Breekel, and Matthew Thompson. Relay cost bounding for contactless EMV payments. In Rainer Böhme and Tatsuaki Okamoto, editors, *Financial Cryptography and Data*

Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers, volume 8975 of *Lecture Notes in Computer Science*, pages 189–206. Springer, 2015.

9. Cas J. F. Cremers, Kasper Bonne Rasmussen, Benedikt Schmidt, and Srdjan Čapkun. Distance hijacking attacks on distance bounding protocols. In *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*, pages 113–127. IEEE Computer Society, 2012.
10. Alexandre Debant and Stéphanie Delaune. Symbolic verification of distance bounding protocols. In Flemming Nielson and David Sands, editors, *Principles of Security and Trust - 8th International Conference, POST 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings*, volume 11426 of *Lecture Notes in Computer Science*, pages 149–174. Springer, 2019.
11. Yvo Desmedt. Major security problems with the 'unforgeable' (feige)-fiat-shamir proofs of identity and how to overcome them. In *SECURICOM'88*, pages 15–17, 1988.
12. Daniel Dolev and Andrew Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.
13. Nancy Durgin, Patrick Lincoln, John Mitchell, and Andre Scedrov. Multiset rewriting and the complexity of bounded security protocols. *Journal of Computer Security*, 12(2):247–311, 2004. Initial version appeared in *Workshop on Formal Methods and Security Protocols*, 1999.
14. Joshua D. Guttman. Shapes: Surveying crypto protocol runs. In Veronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, Cryptology and Information Security Series. IOS Press, 2011.
15. Joshua D. Guttman. Establishing and preserving protocol security goals. *Journal of Computer Security*, 22(2):201–267, 2014.
16. Gerhard P. Hancke and Markus G. Kuhn. An RFID distance bounding protocol. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SecureComm 2005, Athens, Greece, 5-9 September, 2005*, pages 67–73. IEEE, 2005.
17. Max I. Kanovich, Tajana Ban Kirigin, Vivek Nigam, Andre Scedrov, and Carolyn L. Talcott. Timed multiset rewriting and the verification of time-sensitive distributed systems. In Martin Fränzle and Nicolas Markey, editors, *Formal Modeling and Analysis of Timed Systems - 14th International Conference, FORMATS 2016, Quebec, QC, Canada, August 24-26, 2016, Proceedings*, volume 9884 of *Lecture Notes in Computer Science*, pages 228–244. Springer, 2016.
18. Max I. Kanovich, Tajana Ban Kirigin, Vivek Nigam, Andre Scedrov, and Carolyn L. Talcott. Time, computational complexity, and probability in the analysis of distance-bounding protocols. *Journal of Computer Security*, 25(6):585–630, 2017.
19. Chong Hee Kim and Gildas Avoine. RFID distance bounding protocol with mixed challenges to prevent relay attacks. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *Cryptology and Network Security, 8th International Conference, CANS 2009, Kanazawa, Japan, December 12-14, 2009. Proceedings*, volume 5888 of *Lecture Notes in Computer Science*, pages 119–133. Springer, 2009.
20. Chong Hee Kim, Gildas Avoine, François Koeune, François-Xavier Standaert, and Olivier Pereira. The swiss-knife RFID distance bounding protocol. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers*, volume 5461 of *Lecture Notes in Computer Science*, pages 98–115. Springer, 2008.

21. Ueli M. Maurer and Pierre E. Schmid. A calculus for security bootstrapping in distributed systems. *Journal of Computer Security*, 4(1):55–80, 1996.
22. Sjouke Mauw, Zach Smith, Jorge Toro-Pozo, and Rolando Trujillo-Rasua. Distance-bounding protocols: Verification without time and location. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 549–566. IEEE Computer Society, 2018.
23. Sjouke Mauw, Zach Smith, Jorge Toro-Pozo, and Rolando Trujillo-Rasua. Post-collusion security and distance bounding. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 941–958. ACM, 2019.
24. Catherine A. Meadows, Radha Poovendran, Dusko Pavlovic, LiWu Chang, and Paul F. Syverson. Distance bounding protocols: Authentication logic analysis and collusion attacks. In Radha Poovendran, Sumit Roy, and Cliff Wang, editors, *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, volume 30 of *Advances in Information Security*, pages 279–298. Springer, 2007.
25. Simon Meier, Benedikt Schmidt, Cas Cremers, and David A. Basin. The tamarin prover for the symbolic analysis of security protocols. In *Computer Aided Verification (CAV)*, pages 696–701, 2013.
26. Jorge Munilla and Alberto Peinado. Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels. *Wireless Communications and Mobile Computing*, 8(9):1227–1232, 2008.
27. John D. Ramsdell. Deducing security goals from shape analysis sentences. The MITRE Corporation, April 2012. `http://arxiv.org/abs/1204.0480`.
28. John D. Ramsdell and Joshua D. Guttman. CPSA4: A cryptographic protocol shapes analyzer, 2017. `https://github.com/mitre/cpsaexp`.
29. John D. Ramsdell, Joshua D. Guttman, Moses D. Liskov, and Paul D. Rowe. *The CPSA Specification: A Reduction System for Searching for Shapes in Cryptographic Protocols*. The MITRE Corporation, 2009. In `http://hackage.haskell.org/package/cpsa` source distribution, `doc` directory.
30. Kasper Bonne Rasmussen and Srdjan Capkun. Realization of RF distance bounding. In *19th USENIX Security Symposium, Washington, DC, USA, August 11-13, 2010, Proceedings*, pages 389–402. USENIX Association, 2010.
31. Jason Reid, Juan Manuel González Nieto, Tee Tang, and Bouchra Senadji. Detecting relay attacks with timing-based protocols. In Feng Bao and Steven Miller, editors, *Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, Singapore, March 20-22, 2007*, pages 204–213. ACM, 2007.
32. Paul D. Rowe, Joshua D. Guttman, and Moses D. Liskov. Measuring protocol strength with security goals. *International Journal of Information Security*, 15(6):575–596, November 2016. DOI 10.1007/s10207-016-0319-z, `http://web.cs.wpi.edu/~guttman/pubs/ijis_measuring-security.pdf`.
33. F. Javier Thayer, Vipin Swarup, and Joshua D. Guttman. Metric strand spaces for locale authentication protocols. In *Trust Management, IFIP WG 11.11*, volume 321 of *IFIP Advances in Information and Communication Technology*, pages 79–94. Springer, 2010.