Skeletons, Homomorphisms, and Shapes: Characterizing Protocol Executions*

Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer

The MITRE Corporation
shaddin, guttman, jt@mitre.org

1 Introduction

Most protocol analysis tools and techniques operate by proving/disproving security properties of a protocol formulated as predicates in a specific logic. Starting from some initial assumptions, theorem proving or model checking (such as in [8]) techniques can be used to check if a certain security property follows. In this paper, we take a different approach to this problem.

Instead of checking each security property individually, our approach is to characterize all protocol executions compatible with the initial assumptions. The resulting *characterization* is a set of protocol runs that is representative of all possible protocol runs. Some advantages of this approach are:

- We find that the proofs for different security properties often duplicate the same work. With this approach, all the work needed to characterize possible runs can be done once. Then it is easy for a human analyst or simple tool to "read off" the value of any security predicate from the characterization by evaluating that predicate on each bundle in the characterization.
- Security properties of the protocol that were not anticipated by the designer will become apparant.
- The analyst interested in a certain security property can see all the possible attacks/counterexamples as opposed to a single attack.
- Since all possible protocol runs are represented, this gives the protocol designer more insight into the effect of different construction primitives used in the protocol.

In this paper, we will present a framework, based on strand spaces, for analyzing protocols and characterizing their executions. While a generalized notion of our *characterizations* can capture both authentication and secrecy properties, we will restrict this discussion to a simpler notion that exclusively capture authentication properties. We will discuss the applicability to secrecy in future work. Indeed, any algorithm for constructing characterizations must reason about both secrecy and authentication.

As motivation this framework, consider a protocol analyst presented with some initial assumptions about a protocol run. Often this is a single strand

* Supported by the National Security Agency and by MITRE-Sponsored Research.

with some secrecy/freshness assumptions. The analyst can then repeatedly apply inference rules such as the authentication tests [3] in order to infer more about the structure of the protocol run.

At any point in the analysis, the analyst is in possession of some partial information about the structure of the protocol runs possible. We will represent this partial information as a structure we will call a *skeleton*. We will also define information preserving *homomorphisms* between skeletons. Thus, much of protocol analysis can be expressed in terms of skeletons and homomorphisms between them.

While we will relegate discussion of the actual algorithms used to construct these characterizations to future work, we will define what we believe the result should be. Given a set of initial assumptions (an initial skeleton), we will define how a set of protocol runs can *characterize* all possible runs. Furthermore, we will show that there is a minimum such *characterization* : the set of *shapes*.

2 Background

Terms form a free algebra A, built from atomic terms via constructors. The atomic terms are partitioned into the types *principals*, *texts*, *keys*, and *nonces*. An inverse operator is defined on keys. There may be additional operations on atoms, such as an injective *public key of* function or an injective *long term shared key of* function mapping principals to keys. Atoms serve as indeterminates (variables), and are written in italics (e.g. a, N_a, K^{-1}). We assume A contains infinitely many atoms of each type.

Terms in A are freely built from atoms using *tagged concatenation* and *encryption*. The tags are chosen from a set of constants written in sans serif font (e.g. tag). The tagged concatenation using tag of t_0 and t_1 is written $\mathsf{tag} \mathbin{\hat{}} t_0 \mathbin{\hat{}} t_1$. Tagged concatenation using the distinguished tag null of t_0 and t_1 is written $t_0 \mathbin{\hat{}} t_1$. Encryption takes a term t and an atomic key K, and yields a term as result written $\{|t|\}_K$. Fix an A. *Replacements* have only atoms in their range:

Definition 1 (Replacement, Application). A *replacement* is a function α mapping atoms to atoms, such that (1) for every atom a, $\alpha(a)$ is an atom of the same type as a, and (2) α is a homomorphism with respect to the operations on atoms, e.g. in the case of inverse keys, for every key K, $K^{-1} \cdot \alpha = (K \cdot \alpha)^{-1}$.

The *application* of α to t, written $t \cdot \alpha$, homomorphically extends α's action on atoms. More explicitly, if $t = a$ is an atom, then $a \cdot \alpha = \alpha(a)$; and:

$$(\mathsf{tag} \mathbin{\hat{}} t_0 \mathbin{\hat{}} t_1) \cdot \alpha = \mathsf{tag} \mathbin{\hat{}} (t_0 \cdot \alpha) \mathbin{\hat{}} (t_1 \cdot \alpha)$$
$$(\{|t|\}_K) \cdot \alpha = \{|t \cdot \alpha|\}_{K \cdot \alpha}$$

Application distributes through pairing and sets. Thus, $(x, y) \cdot \alpha = (x \cdot \alpha, y \cdot \alpha)$, and $S \cdot \alpha = \{x \cdot \alpha : x \in S\}$. If $x \notin \mathsf{A}$ is a simple value such as an integer or a symbol, then $x \cdot \alpha = x$.

Since replacements map atoms to atoms, not to compound terms, unification is very simple. Two terms are unifiable if and only if they have the same abstract syntax tree structure, with the same tags associated with corresponding concatenations, and the same type for atoms at corresponding leaves. To unify t_1, t_2 means to partition the atoms at the leaves; a most general unifier is a finest partition that maps a, b to the same c whenever a appears at the end of a path in t_1 and b appears at the end of the same path in t_2. If two terms t_1, t_2 are unifiable, then $t_1 \cdot \alpha$ and $t_2 \cdot \beta$ are unifiable.

The direction + means transmission, and the direction − means reception:

Definition 2 (Strand Spaces). A *direction* is one of the symbols $+, -$. A *directed term* is a pair (d, t) with $t \in \mathsf{A}$ and d a direction, normally written $+t, -t$. $(\pm\mathsf{A})^*$ is the set of finite sequences of directed terms.

A *strand space* over A is a structure containing a set Σ and two mappings: a trace mapping $\mathsf{tr} : \Sigma \to (\pm\mathsf{A})^*$ and a replacement application operator $(s, \alpha) \mapsto s \cdot \alpha$ such that (1) $\mathsf{tr}(s \cdot \alpha) = (\mathsf{tr}(s)) \cdot \alpha$, and (2) $s \cdot \alpha = s' \cdot \alpha$ implies $s = s'$.

By condition (2), Σ has infinitely many copies of each strand s, i.e. strands s' with $\mathsf{tr}(s') = \mathsf{tr}(s)$.

Definition 3. A *penetrator strand* has trace of one of the following forms:

M: $\langle +t \rangle$ where $t \in$text, principal,nonce
K: $\langle +K \rangle$
C: $\langle -g,\ -h,\ +g \mathbin{\hat{}} h \rangle$
S: $\langle -g \mathbin{\hat{}} h,\ +g,\ +h \rangle$
E: $\langle -K,\ -h,\ +\{\!|h|\!\}_K \rangle$
D: $\langle -K^{-1},\ -\{\!|h|\!\}_K,\ +h \rangle$.

If s is a penetrator strand, then $s \cdot \alpha$ is a penetrator strand of the same kind.

Definition 4 (Protocols). A *protocol* $\langle \Pi, n, u \rangle$ consists of (1) a finite set of strands called the *roles* of the protocol, and (2) for each role $r \in \Pi$, two sets of atoms n_r, u_r giving *origination data* for r. The *regular strands* Σ_Π *over* Π consists of all instances $r \cdot \alpha$ for $r \in \Pi$.

A *node* is a pair $n = (s, i)$ where $i \leq \mathsf{length}(\mathsf{tr}(s))$; $\mathsf{strand}(s, i) = s$; and the *direction* and *term* of n are those of $\mathsf{tr}(s)(i)$. We prefer to write $s \downarrow i$ for the node $n = (s, i)$. The set $\mathcal{N}$ of all nodes forms a directed graph $\mathcal{G} = \langle \mathcal{N}, (\rightarrow \cup \Rightarrow) \rangle$ with edges $n_1 \rightarrow n_2$ for communication (with the same term, directed from positive to negative node) and $n_1 \Rightarrow n_2$ for succession on the same strand.

The *subterm* relation, written $\sqsubset$, is the least reflexive, transitive relation such that (1) $t_0 \sqsubset \mathsf{tag} \mathbin{\hat{}} t_0 \mathbin{\hat{}} t_1$; (2) $t_1 \sqsubset \mathsf{tag} \mathbin{\hat{}} t_0 \mathbin{\hat{}} t_1$; and (3) $t \sqsubset \{\!|t|\!\}_K$. Notice, however, $K \not\sqsubset \{\!|t|\!\}_K$ unless (anomalously) $K \sqsubset t$. We say that a key K is *used for encryption* in a term t if for some t_0, $\{\!|t_0|\!\}_K \sqsubset t$.

A term t *originates* at node n if n is positive, $t \sqsubset \mathrm{term}(n)$, and $t \not\sqsubset \mathrm{term}(m)$ whenever $m \Rightarrow^+ n$. Thus, t originates on n if t is part of a message transmitted on n, and t was neither sent nor received previously on this strand.

Definition 5 (Bundle). A finite acyclic subgraph $\mathcal{B} = \langle \mathcal{N}_\mathcal{B}, (\rightarrow_\mathcal{B} \cup \Rightarrow_\mathcal{B}) \rangle$ of $\mathcal{G}$ is a *bundle* if (1) if $n_2 \in \mathcal{N}_\mathcal{B}$ is negative, then there is a unique $n_1 \in \mathcal{N}_\mathcal{B}$ with $n_1 \rightarrow_\mathcal{B} n_2$; and (2) if $n_2 \in \mathcal{N}_\mathcal{B}$ and $n_1 \Rightarrow n_2$, then $n_1 \Rightarrow_\mathcal{B} n_2$. When $\mathcal{B}$ is a bundle, $\preceq_\mathcal{B}$ is the reflexive, transitive closure of $(\rightarrow_\mathcal{B} \cup \Rightarrow_\mathcal{B})$.

A bundle $\mathcal{B}$ is *over* $\langle \Pi, n, u \rangle$ if for every $s \downarrow i \in \mathcal{B}$, (1) either $s \in \Sigma_\Pi$ or s is a penetrator strand; (2) if $s = r \cdot \alpha$ and $a \in n_r \cdot \alpha$, then a does not originate in $\mathcal{B}$; and (3) if $s = r \cdot \alpha$ and $a \in u_r \cdot \alpha$, then a originates at most once in $\mathcal{B}$.

Proposition 1. *Let $\mathcal{B}$ be a bundle. $\preceq_\mathcal{B}$ is a well-founded partial order. Every non-empty set of nodes of $\mathcal{B}$ has $\preceq_\mathcal{B}$-minimal members. If α is a replacement, then $\mathcal{B} \cdot \alpha$ is a bundle.*

We will also define the sub-bundle relation between bundles.

Definition 6. A bundle $\mathcal{B}_1$ is a sub-bundle of bundle $\mathcal{B}_2$ is $\mathcal{B}_1$ is a subgraph of $\mathcal{B}_2$, up to (injective) renaming of nodes.

3 Skeletons and Homomorphisms

In this section we will define the framework of skeletons and homomorphisms.

A preskeleton describes the regular (honest) parts of a set of bundles. K is *used* in t if, for some t_0, $\{|t_0|\}_K \sqsubset t$. If a occurs in t or is used in t, then a is *mentioned* in t.

Definition 7. A four-tuple $\mathbb{A} = (\mathsf{node}, \preceq, \mathsf{non}, \mathsf{unique})$ is a *preskeleton* if:

1. node is a finite set of regular nodes; $n_1 \in \mathsf{node}$ and $n_0 \Rightarrow^+ n_1$ implies $n_0 \in \mathsf{node}$;
2. $\preceq$ is a partial ordering on node such that $n_0 \Rightarrow^+ n_1$ implies $n_0 \preceq n_1$;
3. non is a set of keys where if $K \in \mathsf{non}$, then for all $n \in \mathsf{node}$, $K \not\sqsubset \mathrm{term}(n)$, and for some $n' \in \mathsf{node}$, either K or K^{-1} is used in $\mathrm{term}(n')$;
4. unique is a set of atoms where if $a \in \mathsf{unique}$, for some $n \in \mathsf{node}$, $a \sqsubset \mathrm{term}(n)$.

A preskeleton $\mathbb{A}$ is a *skeleton* if in addition:

$4'$. $a \in \mathsf{unique}$ implies a originates at no more than one $n \in \mathsf{node}$.

A skeleton is similar to the notion of semi-bundle in [8].

We select components of a preskeleton using subscripts. For instance, if $\mathbb{A} = (\mathsf{node}, R, S, S')$, then $\preceq_\mathbb{A}$ means R and $\mathsf{unique}_\mathbb{A}$ means S'. We write $n \in \mathbb{A}$ to mean $n \in \mathsf{node}_\mathbb{A}$, and we say that a strand s is in $\mathbb{A}$ when at least one node of s is in $\mathbb{A}$. The $\mathbb{A}$-height of s is the number of nodes of s in $\mathbb{A}$. By Clauses 3 and 4, $\mathsf{unique}_\mathbb{A} \cap \mathsf{non}_\mathbb{A} = \emptyset$.

We will define the sub-[pre]skeleton relation analogously to the sub-bundle relation:

Definition 8. A [pre]skeleton $\mathbb{A}$ is a sub-[pre]skeleton of $\mathbb{A}'$ if each of $\mathsf{node}_\mathbb{A}, \preceq_\mathbb{A}, \mathsf{non}_\mathbb{A}, \mathsf{unique}_\mathbb{A}$ is a subset of $\mathsf{node}_{\mathbb{A}'}, \preceq_{\mathbb{A}'}, \mathsf{non}_{\mathbb{A}'}, \mathsf{unique}_{\mathbb{A}'}$ respectively, up to (injective) renaming of nodes.

Bundles correspond to certain skeletons:

Definition 9. Bundle $\mathcal{B}$ *realizes* skeleton $\mathbb{A}$ if (1) the nodes of $\mathbb{A}$ are precisely the regular nodes of $\mathcal{B}$; (2) $n \preceq_{\mathbb{A}} n'$ just in case $n, n' \in \mathsf{node}_{\mathbb{A}}$ and $n \preceq_{\mathcal{B}} n'$; (3) $K \in \mathsf{non}_{\mathbb{A}}$ just in case $K \not\sqsubseteq \mathrm{term}(n)$ for any $n \in \mathcal{B}$ but K or K^{-1} is used in some $n' \in \mathcal{B}$; (4) $a \in \mathsf{unique}_{\mathbb{A}}$ just in case a originates uniquely in $\mathcal{B}$. if some $\mathcal{B}$ realizes $\mathbb{A}$ we say that $\mathbb{A}$ is a *realized skeleton*

In fact, a bundle completely determines the skeleton it realizes.

Proposition 2. *If $\mathcal{B}$ is a bundle, then there is a unique skeleton that it realizes. By condition (4), $\mathcal{B}$ does not realize $\mathbb{A}$ if $\mathbb{A}$ is a preskeleton but not a skeleton.*

Homomorphisms. Since preskeletons represent partial-information about a protocol run, it would be useful to define information-preserving maps between them: *homomorphisms*

Definition 10. Let $\mathbb{A}_0, \mathbb{A}_1$ be preskeletons, α a replacement, $\phi\colon \mathsf{node}_{\mathbb{A}_0} \to \mathsf{node}_{\mathbb{A}_1}$. $H = [\phi, \alpha]$ is a *homomorphism* if

1a. For all $n \in \mathbb{A}_0$, $\mathrm{term}(\phi(n)) = \mathrm{term}(n) \cdot \alpha$;
1b. For all s, i, if $s \downarrow i \in \mathbb{A}$ then there is an s' s.t. for all $j \leq i$, $\phi(s \downarrow j) = (s', j)$;
2. $n \preceq_{\mathbb{A}_0} m$ implies $\phi(n) \preceq_{\mathbb{A}_1} \phi(m)$;
3. $\mathsf{non}_{\mathbb{A}_0} \cdot \alpha \subset \mathsf{non}_{\mathbb{A}_1}$;
4. $\mathsf{unique}_{\mathbb{A}_0} \cdot \alpha \subset \mathsf{unique}_{\mathbb{A}_1}$.

We write $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ when H is a homomorphism from $\mathbb{A}_0$ to $\mathbb{A}_1$.

Homomorphisms can transform preskeletons to skeletons by contracting nodes that originate the same $a \in \mathsf{unique}_{\mathbb{A}}$. Furthermore, homomorphisms can transform preskeletons to realized skeletons.

There are many runs of the protocol compatible with a set of starting assumptions represented by a preskeleton $\mathbb{A}$. Homomorphisms from $\mathbb{A}$ to realized skeletons describe how $\mathbb{A}$ is part of a protocol run.

Definition 11. We say a homomorphism H realizes preskeleton $\mathbb{A}$ if it maps $\mathbb{A}$ to a realized skeleton $\mathbb{A}'$. In this case, we say $\mathbb{A}$ is *realizable*. Realized skeletons are preskeletons that are realizable via the identity homomorphism.

4 Degeneracy

We use the notion of unique-origination to represent fresh generation of values such as nonces and session keys. This necessitates that we restrict the set of homomorphisms we are interested in to homomorphisms that respect this intended real-world meaning of unique-origination.

Definition 12 (Degenerate Homomorphism). A replacement α is *degenerate for* $\mathbb{A}$ if there are distinct atoms a, b and a strand s where (1) $a \in \mathsf{unique}_{\mathbb{A}}$ originates at $s \downarrow i$ in $\mathbb{A}$, (2) b occurs on $s \downarrow j$ for $j \leq i$, and (3) $a \cdot \alpha = b \cdot \alpha$.

$H = [\phi, \alpha]\colon \mathbb{A}_0 \mapsto \mathbb{A}$ is *degenerate* if α is degenerate for $\mathbb{A}_0$.

A degenerate replacement identifies a uniquely originating atom with some other atom already known at the time it is chosen. Degenerate homomorphisms are of negligible probability relative to stochastic models for protocols [4]. In the rest of this paper, the reader can assume every reference to homomorphisms is in fact restricted to non-degenerate homomorphisms.

In the same spirit, we are only interested in [pre]skeletons/bundles that respect the real-world meaning of unique-origination.

Definition 13 (Degenerate Preskeleton/Bundle). A [pre]skeleton/bundle is degenerate if it contains a strand $r \cdot \alpha$ of role r such that (1) $a \in u_r$ originates at $r \downarrow i$ (2) b occurs on $r \downarrow j$ for $j \leq i$, and (3) $a \cdot \alpha = b \cdot \alpha$.

Likewise, the reader can assume that every mention of [pre]skeletons/bundles is restricted to non-degenerate [pre]skeletons/bundles.

5 Shapes

Due to the existance of an infinite set of homomorphisms realizing any preskeleton $\mathbb{A}$, it is useful to find a small subset of those that characterizes all the runs. The notion of characterization is catered to an intuitive understanding of what it means for a protocol run to be an *extension* of another protocol run. Here we capture one notion of extension that we have, in our experience, found to be the most natural and useful.

Definition 14. A bundle $\mathcal{B}_2$ *extends* a bundle $\mathcal{B}_1$ if there is an atom replacement (note: not necessarily injective) α such that $\mathcal{B}_1 \cdot \alpha$ is a sub-bundle of $\mathcal{B}_2$. Likewise, for [pre]skeletons: A [pre]skeleton $\mathbb{A}_2$ *extends* a [pre]skeleton $\mathbb{A}_1$ if there is a replacement α such that $\mathbb{A}_1 \cdot \alpha$ is a sub-[pre]skeleton of $\mathbb{A}_2$. Naturally extending this to homomorphisms: A homomorphism $H_2 : \mathbb{A} \rightarrow \mathbb{A}_2$ *extends* a homomorphism $H_1 : \mathbb{A} \rightarrow \mathbb{A}_1$ if there is an atom replacement α such that $\alpha \circ H_1 : \mathbb{A} \rightarrow \mathbb{A}_1 \cdot \alpha$ is simply a codomain restriction of H_2. (i.e. $\mathbb{A}_1 \cdot \alpha$ is a sub-[pre]skeleton of $\mathbb{A}_2$.)

This notion of extension can be equivalently, and more formally, stated as follows:

Proposition 3. *[Pre]Skeleton $\mathbb{A}_2$ extends [pre]skeleton $\mathbb{A}_1$ if and only if there is a node-injective homomorphism from $\mathbb{A}_1$ to $\mathbb{A}_2$.*

Homomorphism H_2 extends homomorphism H_1 iff $H_2 = G \circ H_1$ where G is node-injective.

Proof. By definition of *sub-[pre]skeleton* and *extends* □

Now we can define what it means for a set of protocol runs (realized homomorphisms/skeletons) to be characterization for $\mathbb{A}$.

Definition 15. A set X of homomorphisms is a characterization for $\mathbb{A}$ if

- Each $H \in X$ realizes $\mathbb{A}$

- Every homomorphism H' realizing $\mathbb{A}$ extends some $H \in X$

Clearly the set of all realizing homomorphisms for $\mathbb{A}$ is a characterization for $\mathbb{A}$. However, we seek a characterization as small as possible: a *minimal characterization*, under some criterion of size (number of nodes, total size of messages... etc). We will show later that there is a a single characterization that is a unique minimum for all these definitions of size simultaneously.

This notion of extension described above captures what it means for one protocol run to simply be an elaboration on another run. One can make an argument for several alternative notions of *extends*, including not requiring node-injectivity for G in 3 above, and/or requiring atom-injectivity of G. Any of these definitions would yield characterizations that can be used to decide security predicates. However, they differ in their size, the ease by which they can be algorithmically constructed, and the ease by which an analyst can interpret the set of all runs and make judgements based on them. We chose this notion of *extends* based on our experience in automating protocol analysis, and we found that the minimum characterizations it yields succinctly capture the intuitive notion of "representative set of all protocol runs".

Requiring node-injectivity in our definition of *extends* has another advantage, in that it results in a partial order as opposed to a preorder. Furthermore, the partial order is well-founded.

Definition 16. For skeletons, $\mathbb{A}_1 \sqsubseteq \mathbb{A}_2$ iff $\mathbb{A}_2$ extends $\mathbb{A}_1$. Likewise for homomorphisms, $H_1 \sqsubseteq H_2$ iff H_2 extends H_1.

Proposition 4. *$\sqsubseteq$ is a well-founded partial order on skeletons (up to isomomorphism).*

Proof. Clearly $\sqsubseteq$ is reflexive and transitive.
Antisymmetry: if $\mathbb{A} \sqsubseteq \mathbb{A}'$ via $H = [\phi, \alpha]$ and $\mathbb{A}' \sqsubseteq \mathbb{A}$ via H' then both have the same number of nodes. By considering th composition of H and H', we can see that H is node-bijective (ϕ is bijective) and atom-bijective (α is bijective). By considering the composition of H and H', we can also see that α is bijective from $\mathsf{non}_{\mathbb{A}}$ to $\mathsf{non}_{\mathbb{A}'}$, and also from $\mathsf{unique}_{\mathbb{A}}$ to $\mathsf{unique}_{\mathbb{A}'}$. Likewise ϕ is a bijection from the pairs in $\preceq_{\mathbb{A}}$ to $\preceq_{\mathbb{A}'}$. Therefore $H^{-1} = [\phi^{-1}, \alpha{-}1]$ is a homomorphism, and is the inverse of H. H is an isomomorphism.
Well-foundedness: For a skeleton $\mathbb{A}$, let $Occ(\mathbb{A})$ be the number of atom occurences in $\mathbb{A}$. Let $Atoms(\mathbb{A})$ be the number of distinct atoms in $\mathbb{A}$, and define the atom-redundancy of a skeleton as $red(\mathbb{A}) = Occ(\mathbb{A}) - Atoms(\mathbb{A})$. Also, let $nonOcc(\mathbb{A})$ and $uniqOcc(\mathbb{A})$ be the number of occurences in $\mathbb{A}$ of atoms in $\mathsf{non}_{\mathbb{A}}$ and $\mathsf{unique}_{\mathbb{A}}$, respectively. We can show that $|\mathsf{node}|$, red , $nonOcc$, $uniqOcc$ and $|\preceq|$ are each non-decreasing with $\sqsubseteq$ and lower bounded by 0. Let

$$Rank(\mathbb{A}) = |\mathsf{node}_{\mathbb{A}}| + red(\mathbb{A}) + nonOcc(\mathbb{A}) + uniqOcc(\mathbb{A}) + |\preceq_{\mathbb{A}}|$$

The rank of a skeleton is non-decreasing with $\sqsubseteq$ and lower bounded by 0. Furthermore, we can show that if $\mathbb{A} \sqsubseteq \mathbb{A}'$, and $Rank(\mathbb{A}) = Rank(\mathbb{A}')$, then $\mathbb{A}$ and $\mathbb{A}'$ are isomorphic. It follows that $\sqsubseteq$ is well-founded up to isomorphism.

□

This extends analogously to homomorphisms.

Corollary 5. *$\sqsubseteq$ is a well-founded partial order on homomorphisms (up to isomorphism)*

We will define shapes as the minimal realizing homomorphisms in this well-founded partial order

Definition 17. A homomorphism H is a *shape* for $\mathbb{A}$ if H realizes $\mathbb{A}$ and H is minimal under $\sqsubseteq$ amongst homomorphisms realizing $\mathbb{A}$. Let $shapes(\mathbb{A})$ be the set of distinct (up to isomorphism) shapes of $\mathbb{A}$.

Next, it becomes apparent that that the set of shapes of $\mathbb{A}$ describes (can be extended to) all runs compatible with $\mathbb{A}$

Proposition 6. *$shapes(\mathbb{A})$ is a characterization for $\mathbb{A}$*

Proof. Since $\sqsubseteq$ is well-founded, for any H' realizing $\mathbb{A}$ there is an $H \in shapes(\mathbb{A})$ such that $H \sqsubseteq H'$, and H' extends H. □

In fact, the set of shapes is the minimum such set

Proposition 7. *$shapes(\mathbb{A})$ is a subset of any other characterization for $\mathbb{A}$. Hence $shapes(\mathbb{A})$ is the minimum characterization for $\mathbb{A}$.*

Proof. Take any other characterization X for $\mathbb{A}$. Take any shape $H \in shapes(\mathbb{A})$. Since X is a characterization, there must be some $G \in X$ such that $G \sqsubseteq H$. Since H is minimal under $\sqsubseteq$ then $G = H$.

□

From 7 we can see that $shapes(\mathbb{A})$ is the smallest characterization for $\mathbb{A}$ under most definitions of size (number of nodes, total length of messages... etc).

6 Examples

6.1 An ISO Candidate

The protocol shown in Figure 1 was a candidate considered by an ISO committee as a *pure* authentication protocol [2]. No shared secret is achieved. It was intended merely to assure each principal that the expected partner was initiating communications.

This protocol was rejected by the committee due to the discovery of an attack [2], shown in Figure 2. Since it was discovered by the Canadian representatives to the committee, it is sometimes called the Canadian attack [2]. The attacker is denoted by P.

This attack constitutes a failure of authentication from the perspective of responder B. Even if both B and A had uncompromised private keys, the protocol does not guarantee to the responder B that A is running a corresponding instance of the initiator role. In this attack, A is running an instance of the responder role.

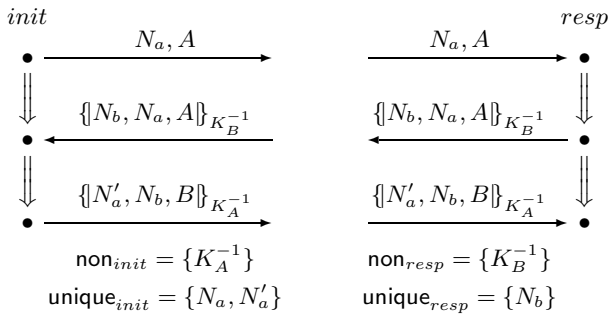

Fig. 1. An ISO Candidate Protocol

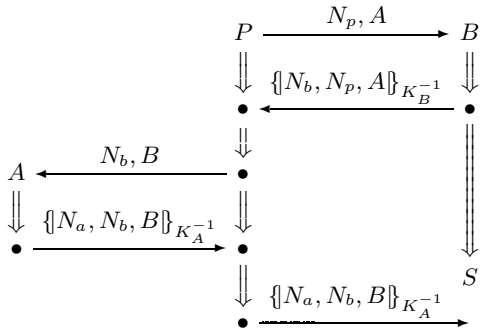

Fig. 2. Bundle: The Canadian Attack

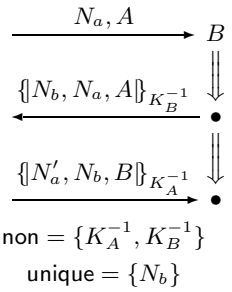

Fig. 3. ISO Candidate Responder Skeleton

To express the guarantees this protocol makes to the responder, we will use the skeleton in Figure 3 as our starting set of assumptions.

It can be shown that there are two shapes for the above skeleton, one corresponding to the intended run, and one corresponding to the attack. These shapes are shown in Figures 4 and 5.

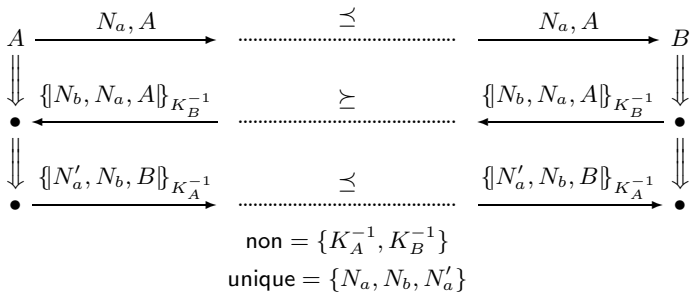

Fig. 4. ISO Candidate Shape 1 for Responder : Intended Run

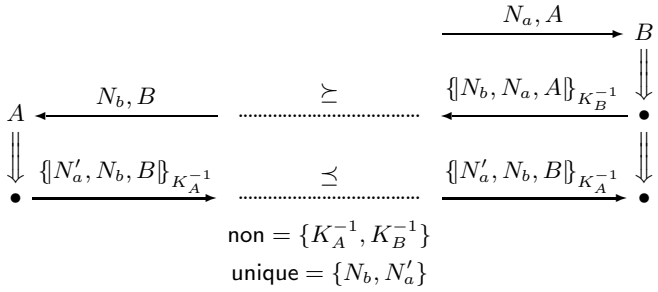

Fig. 5. ISO Candidate Shape 2 for Responder : Attack

6.2 Needham Schroeder

The well-known Needham Schroeder protocol [6] shown in Figure 6 also suffers from an attack [5] on authentication from the perspective of the responder. The attack is shown in Figure 7.

We will start with a skeleton of the responder as our starting assumptions. The skeleton is shown in Figure 8.

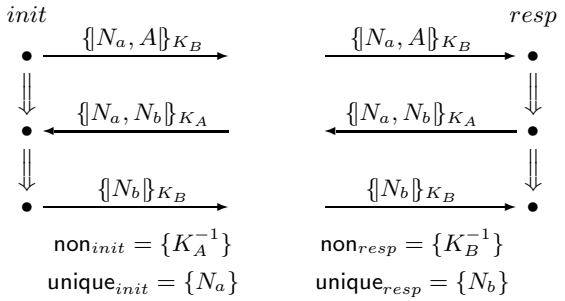

Fig. 6. The Needham Schroeder Protocol

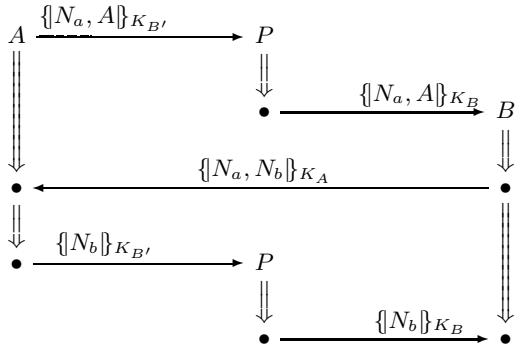

Fig. 7. Bundle: Man-in-the-Middle attack on Needham-Schroeder

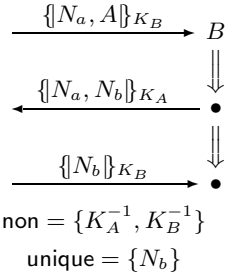

Fig. 8. Needham Shroeder Responder Skeleton

There is a subtlety here. Despite both the intended run and the attack being possible executions, it turns out that the shape in Figure 9 is the only shape for the skeleton in Figure 8.

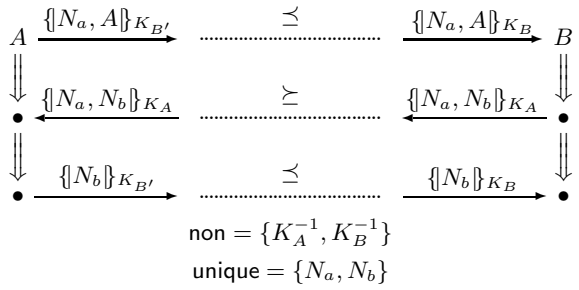

Fig. 9. Only Shape for Needham Schroeder Responder

The reason for this is that the realized skeleton corresponding to the intended run is not minimal under $\sqsubseteq$. It is in fact strictly greater under $\sqsubseteq$ than the realized skeleton in figure 9. A node injective homomorphism identifying B and B' maps the shape to the intended run. Therefore, the intended run is simply an elaboration of the shape in Figure 9.

7 Conclusions and Future Work

In cases where the set of shapes is finite, the shapes for a preskeleton $\mathbb{A}$ form a succint representation of all protocol runs compatible with $\mathbb{A}$. In most protocols we have studied, we have found experimentally that any starting preskeleton yields a finite set of shapes. However, it would be useful to identify a subclass of protocols for which this is guaranteed. We expect that a subclass of protocols similar to those defined in [1] and [7] has this property.

We are also developing an algorithm for constructing the set of shapes for a preskeleton. At a high level, the algorithm starts with the initial preskeleton, finds and solves an authentication test (see [3]) to yield a finite set of solution preskeletons; then recurses on those. This results is a tree of preskeletons with the shapes as the leaves. These shapes are annotated with secrecy information. The algorithm reasons about secrecy by trying to construct a shape that discloses a value. Details of the algorithm and an exploration of its formal properties will be the subject of future work.

References

1. Bruno Blanchet and Andreas Podelski. Verification of cryptographic protocols: Tagging enforces termination. In Andrew D. Gordon, editor, *Foundations of Soft-*

ware Science and Computation Structures, number 2620 in LNCS, pages 136–152. Springer, April 2003.

2. Joshua D. Guttman. Security goals: Packet trajectories and strand spaces. In Roberto Gorrieri and Riccardo Focardi, editors, *Foundations of Security Analysis and Design*, volume 2171 of *LNCS*, pages 197–261. Springer Verlag, 2001.
3. Joshua D. Guttman and F. Javier Thayer. Authentication tests and the structure of bundles. *Theoretical Computer Science*, 283(2):333–380, June 2002.
4. Joshua D. Guttman, F. Javier Thayer, and Lenore D. Zuck. The faithfulness of abstract protocol analysis: Message authentication. *Journal of Computer Security*, 12(6):865–891, 2004.
5. Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proceeedings of* TACAS, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer Verlag, 1996.
6. Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12), 1978.
7. R. Ramanujam and S. P. Suresh. A decidable subclass of unbounded security protocol. In R. Gorrieri, editor, *WITS '03: Workshop on Issues in the Theory of Security*, pages 11–20, Warsaw, April 2003.
8. Dawn Xiaodong Song. Athena: a new efficient automated checker for security protocol analysis. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, June 1999.