Skeletons and the Shapes of Bundles*

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

The MITRE Corporation

Abstract. The *shapes* of a protocol are its minimal, essentially different executions. Naturally occurring protocols have only finitely many, indeed very few shapes. Authentication and secrecy properties are easy to determine from the shapes, as are attacks and anomalies. In this paper, we define the idea of shape, and we also provide some operations that can be used to construct shapes.

These operations are versions of the two *authentication tests*, fundamental patterns for protocol analysis and heuristics for protocol design. The authentication tests were originally presented as theorems about all complete executions. We have strengthened those results here. We also use them to infer construction operations for shapes. These construction operations work on partial descriptions of executions, and serve as information-increasing transformations on the descriptions.

1 The Idea of Shapes

In this paper, we study how to construct the *shapes* of a protocol, where by shapes we mean the minimal, essentially different executions of a protocol. From the shapes, one can read off what exactly secrecy and authentication properties a protocol satisfies, as well as observe other anomalies in possible executions. In this, our approach differs from much work in protocol analysis, which aims at safe approximations (e.g. [5, 1]). We also differ from work using bounded protocol analysis (e.g. [2, 10]); the shapes describe protocol executions of all sizes.

In practice, protocols have remarkably few shapes. The Needham-Schoeder-Lowe [11, 9] protocol has only one. This holds whether we take the point of view of a responder B, asking what global behavior must have occurred if B has had a local run of the protocol, or whether we start from a local run of an originator A. In either case, the other party must have had a matching run. A, however, can never be sure that the last message it sends was received by B, as A is no longer expecting to receive any further messages. Uniqueness of shape is perhaps not surprising for as strong a protocol as Needham-Schroeder-Lowe.

However, even a flawed protocol such as the original Needham-Schroeder protocol may have a unique shape, shown in Fig. 1.

* Supported by the National Security Agency and by MITRE-Sponsored Research.
Addresses: shaddin@stanford.edu, {guttman, jt}@mitre.org.

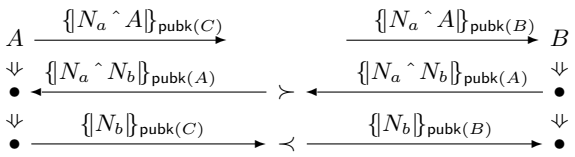

Fig. 1. Needham-Schroeder Shape for B ($\mathsf{privk}(A)$ uncompromised, N_b fresh)

Terminology. Newly introduced terminology is in **boldface**.

B's local behavior is represented by the right-hand column in Fig. 1, consisting of nodes connected by double arrows $\bullet \Rightarrow \bullet$. A's local behavior is represented by the left-hand column. We call such a column a **strand**. The **nodes** represent message transmission or reception events, and the double arrows represent succession within a single linearly ordered local activity. The message transmitted or received on a node n is written $\mathsf{msg}(n)$. A **regular strand** is a strand that represents a principal executing a single local session of a protocol; it is called a regular strand because the behavior follows the protocol rules. A *local behavior* as used so far refers to a regular strand. (See Section 2.2.)

In the messages, we use $\{|t|\}_K$ to refer to the encryption of t with key K, and $t \hat{} t'$ means the pair of the messages t and t'. Messages are constructed freely via these two operations from atomic values such as principal names A, nonces N_a, keys K, etc. (See Section 2.1.)

The **subterm** relation is the least reflexive, transitive relation such that t is a subterm of $\{|t|\}_K$, t is a subterm of $t \hat{} t'$, and t is a subterm of $t' \hat{} t$ (for all K, t'). We write $t \sqsubseteq t'$ if t is a subterm of t'. Thus, $K \not\sqsubseteq \{|t|\}_K$ unless (anomalously) $K \sqsubseteq t$. Instead, K contributed to *how* $\{|t|\}_K$ was produced. This terminology has an advantage: Uncompromised long-term keys are never subterms of messages transmitted in a protocol; they are used by regular principals to encrypt, decrypt, or sign messages, but are never transmitted. A value a **originates at** a node n if (1) n is a transmission node; (2) $a \sqsubseteq \mathsf{msg}(n)$; and (3) if m is any earlier node on the same strand, then $a \not\sqsubseteq \mathsf{msg}(m)$. (Section 2.2, Example 3.)

Adversary behavior is represented by strands too. These **penetrator strands** codify the basic abilities that make up the Dolev-Yao model. They include transmitting a basic value such as a nonce or a key; transmitting an encrypted message after receiving its plaintext and the key; and transmitting a plaintext after receiving ciphertext and decryption key. The adversary can also pair two messages, or separate the pieces of a paired message. Since a penetrator strand that encrypts or decrypts must receive the key as one of its inputs, keys used by the adversary—compromised keys—have always been transmitted by some participant. These penetrator strands are independent of the protocol under analysis. (See Definition 3.)

Suppose that $\mathcal{B}$ is a finite, directed acyclic graph whose nodes lie on regular and penetrator strands, and whose edges are either (a) strand succession edges

$n_0 \Rightarrow n_1$, or else (b) message transmission edges $n \rightarrow m$ where $\mathsf{msg}(n) = \mathsf{msg}(m)$, n is a transmission node, and m is a reception node.

$\mathcal{B}$ is a **bundle** if (1) if $n_0 \Rightarrow n_1$ and $n_1 \in \mathcal{B}$, then $n_0 \in \mathcal{B}$, and (2) for every reception node $m \in \mathcal{B}$, there is a unique transmission node $n \in \mathcal{B}$ such that the edge $n \rightarrow m$ is in $\mathcal{B}$. The conditions (1,2) ensure that $\mathcal{B}$ is causally well founded. A *global behavior* or *execution*, as used so far, refers to a bundle. (See Definition 5.)

The NS Shape. In the Needham-Schroeder protocol, let us suppose that B's nonce N_b has been freshly chosen and A's private key $\mathsf{privk}(A)$ is uncompromised, and that B has executed the strand shown at the right in Fig. 1. In protocols using asymmetric encryption, the private keys are used only by recipients to destructure incoming messages. Given that—on a particular occasion—B received and sent these messages, what must have occurred elsewhere in the network?

A must have had a partially matching strand, with the messages sent and received in the order indicated by the arrows of both kinds and the connecting symbols $\prec$. These symbols mean that the endpoints are ordered, but that other behavior may intervene, whether adversary strands or regular strands. A's strand is only partially matching, because the principal A meant to contact is some C which may or may not equal B. There is no alternative: Any diagram containing the responder strand of Fig. 1 must contain at least an instance of the initiator strand, with the events ordered as shown, or it cannot have happened.

Such a diagram is a *shape*. A shape consists of the regular strands of some execution, forming a *minimal* set containing the initial regular strands (in this case, just the right-hand column). Possible executions may freely add adversary behavior. Each shape is relative to assumptions about keys and freshness, in this case that $\mathsf{privk}(A)$ is uncompromised and N_b freshly chosen.

Although there is a single shape, there are two ways that this shape may be realized in executions. Either (1) C's private key may be compromised, in which case we may complete this diagram with adversary activity to obtain the Lowe attack [9]; or else (2) $C = B$, leading to the intended run.

Some protocols have more than one shape, Otway-Rees, e.g., having four. In searching for shapes, one starts from some initial set of strands. Typically, the initial set is a singleton, which we refer to as the "point of view" of the analysis.

Skeletons, Homomorphisms, Shapes. A *skeleton* represents regular (non-penetrator) behavior that might make up part of an execution, and a *homomorphism* is an information-preserving map between skeletons. Skeletons are partially-ordered structures, like fragments of Lamport diagrams [8] or fragments of message sequence charts [7].

A **skeleton** $\mathbb{A}$ is (1) a finite set of regular nodes, equipped with additional information. The additional information consists of (2) a partial order $\preceq_{\mathbb{A}}$ on the nodes indicating causal precedence; (3) a set of keys $\mathsf{non}_{\mathbb{A}}$; and (4) a set

of atomic values $\mathsf{unique}_{\mathbb{A}}$. Values in $\mathsf{non}_{\mathbb{A}}$ must originate nowhere in $\mathbb{A}$, whereas those in $\mathsf{unique}_{\mathbb{A}}$ originate at most once in $\mathbb{A}$.[1] (See Def. 7.)

$\mathbb{A}$ is **realized** if it has precisely the regular behavior of some execution. Every message received by a regular participant either should have been sent previously, or should be constructable by the adversary using messages sent previously. (See Def. 9.)

Example 1. Fig. 1 shows skeleton $\mathbb{A}_{ns}$, with $\mathsf{non}_{\mathbb{A}_{ns}} = \{\mathsf{privk}(A)\}$ and $\mathsf{unique}_{\mathbb{A}_{ns}} = \{N_b\}$. $\mathbb{A}_{ns}$ is a realized skeleton.

The right-hand strand of Fig. 1, B's responder strand, also forms a skeleton $\mathbb{A}_b$ with the same choice of $\mathsf{non}, \mathsf{unique}$. $\mathbb{A}_b$ is not realized.

The first two nodes on Fig. 1 also form a skeleton $\mathbb{A}_{b_2}$. This skeleton is realized, as the adversary can prepare the incoming message of its first node, and discard the outgoing message of its second node.

The result of replacing C by B throughout $\mathbb{A}_{ns}$—hence replacing $\mathsf{pubk}(C)$ by $\mathsf{pubk}(B)$—yields a realized skeleton $\mathbb{A}_{nsi}$, the Needham-Schroeder intended run.

A **homomorphism** is a map H from $\mathbb{A}_0$ to $\mathbb{A}_1$, written $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$. We represent it as a pair of maps (ϕ, α), where ϕ maps the nodes of $\mathbb{A}_0$ into those of $\mathbb{A}_1$, and α is a **replacement** mapping atomic values into atomic values. We write $t \cdot \alpha$ for the result of applying a replacement α to a message t. $H = (\phi, \alpha)$ is a homomorphism iff: (1) ϕ respects strand structure, and $\mathsf{msg}(n) \cdot \alpha = \mathsf{msg}(\phi(n))$ for all $n \in \mathbb{A}_0$; (2) $m \preceq_{\mathbb{A}_0} n$ implies $\phi(m) \preceq_{\mathbb{A}_1} \phi(n)$; (3) $\mathsf{non}_{\mathbb{A}_0} \cdot \alpha \subseteq \mathsf{non}_{\mathbb{A}_1}$; and (4) $\mathsf{unique}_{\mathbb{A}_0} \cdot \alpha \subseteq \mathsf{unique}_{\mathbb{A}_1}$. (Defs. 1, 11.)

Homomorphisms are *information-preserving* transformations. Each skeleton $\mathbb{A}_0$ describes the realized skeletons reachable from $\mathbb{A}_0$ by homomorphisms. Since homomorphisms compose, if $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ then any realized skeleton accessible from $\mathbb{A}_1$ is accessible from $\mathbb{A}_0$. Thus, $\mathbb{A}_1$ preserves the information in $\mathbb{A}_0$: $\mathbb{A}_1$ describes a subset of the realized skeletons described by $\mathbb{A}_0$.

A homomorphism may supplement the strands of $\mathbb{A}_0$ with additional behavior in $\mathbb{A}_1$; it may affect atomic parameter values; and it may identify different nodes together, if their strands are compatible in messages sent and positions in the partial ordering.

Example 2. The map $H_{ns}\colon \mathbb{A}_b \mapsto \mathbb{A}_{ns}$ embedding the responder strand of Fig. 1 into $\mathbb{A}_{ns}$ is a homomorphism. Likewise if we embed the first two nodes of B's strand (rather than all of $\mathbb{A}_b$) into $\mathbb{A}_{ns}$. Another homomorphism $H_i\colon \mathbb{A}_{ns} \mapsto \mathbb{A}_{nsi}$ rewrites each occurrence of C in $\mathbb{A}_{ns}$ to B, hence each occurrence of $\mathsf{pubk}(C)$ to $\mathsf{pubk}(B)$. It exhibits the Needham-Schroeder intended run as an instance of Fig. 1. The composition $H_{nsi} = H_i \circ H_{ns}$ embeds the responder strand into the intended run.

A homomorphism $H = (\phi, \alpha)$ is **nodewise injective** if the function ϕ on nodes is injective. The nodewise injective homomorphisms determine a useful partial order on homomorphisms: When for some nodewise injective H_1, $H_1 \circ H = H'$, we write $H \leq_n H'$. If $H \leq_n H' \leq_n H$, then H and H' are isomorphic.

[1] When $n \Rightarrow^* n'$ and $n' \in \mathbb{A}$, we require $n \in \mathbb{A}$ and $n \preceq_{\mathbb{A}} n'$.

A homomorphism $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ is a **shape** iff (a) $\mathbb{A}_1$ is realized and (b) H is $\leq_n$-minimal among homomorphisms from $\mathbb{A}_0$ to realized skeletons. If H is a shape, and we can factor H into $\mathbb{A}_0 \overset{H_0}{\mapsto} \mathbb{A}' \overset{H_1}{\mapsto} \mathbb{A}_1$, where $\mathbb{A}'$ is realized, then $\mathbb{A}'$ cannot contain fewer nodes than $\mathbb{A}_1$, or identify fewer atomic values. $\mathbb{A}_1$ is as small and as general as possible. (Def. 13.)

We call a *skeleton* $\mathbb{A}_1$ a shape when the homomorphism H (usually an embedding) is understood. In this looser sense, Fig. 1 shows the shape $\mathbb{A}_{ns}$. Strictly, the embedding $H_{ns}\colon \mathbb{A}_b \mapsto \mathbb{A}_{ns}$ is the shape. The embedding $H_{nsi}\colon \mathbb{A}_b \mapsto \mathbb{A}_{nsi}$, with target the Needham-Schroeder intended run $\mathbb{A}_{nsi}$, is not a shape. $\mathbb{A}_{ns}$ identifies fewer atoms, and the map replacing C with B is a nodewise injective $H_i\colon \mathbb{A}_{ns} \mapsto \mathbb{A}_{nsi}$, so $H_{ns} \leq_n H_i \circ H_{ns} = H_{nsi}$.

Shapes exist below realized skeletons: If $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ with $\mathbb{A}_1$ realized, then the set of shapes H_1 with $H_1 \leq_n H$ is finite and non-empty. (Prop. 7.)

2 Terms, Strands, and Bundles

In this section and Section 4 we give precise definitions, which include a number of fine points which seemed an unnecessary distraction in Section 1. In this section, the definitions of replacement and protocol (Defs. 1, 4) are new versus [6].

2.1 Algebra of Terms

Terms (or messages) form a free algebra A, built from atomic terms via constructors. The atoms are partitioned into the types *principals*, *texts*, *keys*, and *nonces*. An inverse operator is defined on keys. There may be additional functions on atoms, such as an injective *public key of* function $\mathsf{pubk}(a)$ mapping principals to keys, or an injective *long term shared key of* function $\mathsf{ltk}(a)$ mapping pairs of principals to keys. These functions are not constructors, and their results are atoms. $\mathsf{pubk}(a)^{-1}$ is a's private key, where $\mathsf{pubk}(a)^{-1} \neq \mathsf{pubk}(a)$. We often write the public key pair as K_a, K_a^{-1}. By contrast, $\mathsf{ltk}(a)^{-1} = \mathsf{ltk}(a)$.

Atoms, written in italics (e.g. a, N_a, K^{-1}), serve as indeterminates (variables). We assume A contains infinitely many atoms of each type. Terms in A are freely built from atoms using *tagged concatenation* and *encryption*. The tagged concatenation using tag of t_0 and t_1 is written $\mathsf{tag}\,\hat{}\,t_0\,\hat{}\,t_1$. Tagged concatenation using the distinguished tag null of t_0 and t_1 is written $t_0\,\hat{}\,t_1$. Encryption takes a term t and an atomic key K, and yields a term as result written $\{|t|\}_K$.

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, i.e., $\alpha(K^{-1}) = (\alpha(K))^{-1}$ and $\alpha(\mathsf{pubk}(a)) = \mathsf{pubk}(\alpha(a))$.

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:

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

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

2.2 Strands and Origination

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 still 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 (2), Σ has infinitely many copies of each 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_t: $\langle +t \rangle$ where $t \in$text, principal,nonce
K_K: $\langle +K \rangle$
$\mathsf{C}_{g,h}$: $\langle -g,\ -h,\ +g \mathbin{\hat{}} h \rangle$
$\mathsf{S}_{g,h}$: $\langle -g \mathbin{\hat{}} h,\ +g,\ +h \rangle$
$\mathsf{E}_{h,K}$: $\langle -K,\ -h,\ +\{\!|h|\!\}_K \rangle$
$\mathsf{D}_{h,K}$: $\langle -K^{-1},\ -\{\!|h|\!\}_K,\ +h \rangle$.

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

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

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 $n = (s, i)$. A term t *originates* at node n if n is positive, $t \sqsubseteq \mathsf{msg}(n)$, and $t \not\sqsubseteq \mathsf{msg}(m)$ 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. If a originates on strand s, we write $\mathcal{O}(s, a)$ to refer to the node on which it originates.

Example 3. N_a originates on the first node of the Needham-Schroeder initiator strand s_i, so we write $\mathcal{O}(s_i, N_a) = s_i \downarrow 1$. N_b originates on the second node of the responder strand s_r, i.e. $\mathcal{O}(s_r, N_b) = s_r \downarrow 2$. More precisely, $\mathcal{O}(s_r, N_b) = s_r \downarrow 2$ unless $N_b = N_a$, because if the two nonces were the same, then N_b would not

originate on the responder strand at all. Instead, it would have been received before being re-transmitted. Thus, the replacement $\beta = [N_b \mapsto N_a]$ destroys the point of origination. Even if we have $\mathcal{O}(s_r, N_b) = s_r \downarrow 2$, we have $\mathcal{O}(s_r \cdot \beta, N_b \cdot \beta)$ undefined. In this sense, applying β to s_r is a kind of degeneracy that destroys a point of origination. When we have assumed that a value such as N_b originates uniquely, we will avoid applying replacements that would destroy its point of origination. (See Def. 4, regular strands, and Def. 11, homomorphism.)

A *listener role* is a regular strand $\mathsf{Lsn}[a]$ with trace $\langle -a \rangle$. It documents that a is available on its own to the adversary, unprotected by encryption. Since replacements respect type, atoms of different type must be overheard by different roles. We assume each protocol Π has listener roles $\mathsf{Lsn}[N]$ and $\mathsf{Lsn}[K]$ for nonces and keys respectively, with traces $\langle -N \rangle$ and $\langle -K \rangle$.

2.3 Protocols and Bundles

Definition 4 (Protocols). A *candidate* $\langle \Pi, \mathsf{strand\_non}, \mathsf{strand\_unique} \rangle$ consists of: (1) a finite set Π of strands—containing the listener strands $\mathsf{Lsn}[N], \mathsf{Lsn}[K]$—called the *roles* of the protocol; (2) a function $\mathsf{strand\_non}$ mapping each role r to a finite set of keys $\mathsf{strand\_non}_r$, called the non-originating keys of r; and (3) a function $\mathsf{strand\_unique}$ mapping each role r to a finite set of atoms $\mathsf{strand\_unique}_r$ called the uniquely originating atoms of r.

A candidate $\langle \Pi, \mathsf{strand\_non}, \mathsf{strand\_unique} \rangle$ is a *protocol* if (1) $K \in \mathsf{strand\_non}_r$ implies that K does not occur in any node of r, but either K or K^{-1} is used for encryption on some term of $\mathsf{tr}(r)$; and (2) $a \in \mathsf{strand\_unique}_r$ implies that a originates on r, i.e. $\mathcal{O}(r, a)$ is well defined.

The *regular strands* of $\langle \Pi, \mathsf{strand\_non}, \mathsf{strand\_unique} \rangle$ form the set $\Sigma_\Pi =$

$$\{r \cdot \alpha \colon r \in \Pi \text{ and } \forall a \in \mathsf{strand\_unique}_r, (\mathcal{O}(r, a)) \cdot \alpha = \mathcal{O}(r \cdot \alpha, a \cdot \alpha)\}.$$

The non-originating keys $\mathsf{strand\_non}_r$ and uniquely originating atoms $\mathsf{strand\_unique}_r$ are used in Defs. 8 and 14, Clauses 1c,d. The condition that constrains $r \cdot \alpha$ based on $\mathcal{O}(r, a)$ is a non-degeneracy condition. It says that replacement α determines an instance of r only if it does not cause a value a, assumed uniquely originating, to collide with another value already encountered in executing r. Since for $a \in \mathsf{strand\_unique}_r$, the left hand side of $(\mathcal{O}(r, a)) \cdot \alpha = \mathcal{O}(r \cdot \alpha, a \cdot \alpha)$ is well-defined, we interpret the equation as meaning that the right hand side is also well-defined, and has the same value.

Example 4. The Needham-Schroeder protocol has a set Π_{ns} of roles containing the two roles shown in Fig. 1 and two listener roles, to hear nonces and keys. For each $r \in \Pi_{ns}$, $\mathsf{strand\_non}_r = \emptyset = \mathsf{strand\_unique}_r$.

Setting $\mathsf{strand\_non}_{init} = \{\mathsf{privk}(B)\}$, $\mathsf{strand\_non}_{resp} = \{\mathsf{privk}(A)\}$ reproduces the original Needham-Schroeder [11] assumption that each peer chosen is uncompromised. The protocol achieves its goals relative to this assumption.

Setting $\mathsf{strand\_unique}_{init} = \{N_a\}$ would express the assumption that every initiator uses a strong random number generator to select nonces, so that the probability of a collision or of an adversary guessing a nonce is negligible.

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.

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* $\langle \Pi, \mathsf{strand\_non}, \mathsf{strand\_unique} \rangle$ 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 $a \in \mathsf{strand\_non}_r \cdot \alpha$, then a does not occur in $\mathcal{B}$; and (3) if $s = r \cdot \alpha$ and $a \in \mathsf{strand\_unique}_r \cdot \alpha$, then a originates at most once in $\mathcal{B}$.

Example 5. Fig. 1 is a bundle if we replace C with B and then connect arrows with matching labels. Alternatively, it becomes a bundle by adding penetrator strands to unpack values encrypted with K_C and repackage them encrypted with K_B.

We say that a strand s is *in* $\mathcal{B}$ if s has at least one node in $\mathcal{B}$. Henceforth, assume fixed some arbitrary protocol $\langle \Pi, \mathsf{strand\_non}, \mathsf{strand\_unique} \rangle$.

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.*

$\mathcal{B} \cdot \alpha$ is a bundle if, for every regular strand $s = r \cdot \beta$ in $\mathcal{B}$, and for every $a \in \mathsf{strand\_unique}_r \cdot \beta$, we have $(\mathcal{O}(s, a)) \cdot \alpha = \mathcal{O}(s \cdot \alpha, a \cdot \alpha)$.

3 Strengthened Authentication Tests in Bundles

To direct the process of searching for realized skeletons, we use the *authentication tests* [6] in a strengthened and simplified form.

3.1 "Occurs Only Within"

An *outgoing test node* receives a uniquely originating atom in a new form, while an *incoming test node* receives an encryption in a new form. A message t occurs in a new form in $\mathsf{msg}(n)$ if it occurs outside a set S of encryptions, whereas previously t occurred only within members of S:

Definition 6 (Occurs only within/outside). A term t_0 *occurs only within* S in t, where S is a set of encryptions, if:

1. $t_0 \not\sqsubseteq t$; or
2. $t \in S$; or
3. $t \neq t_0$ and either (3a) $t = \{\!|t_1|\!\}_K$ and t_0 occurs only within S in t_1; or (3b) $t = \mathsf{tag} \,\hat{}\, t_1 \,\hat{}\, t_2$ and t_0 occurs only within S in each t_i $(i = 1, 2)$.

It *occurs outside* S *in* t if t_0 does not occur only within S in t.

We say that t *exits* S *passing from* t_0 *to* t_1 if t occurs only within S in t_0 but t occurs outside S in t_1. Term t *exits* S *at* a node n if t occurs outside S in $\mathsf{msg}(n)$ but occurs only within S in every $\mathsf{msg}(m)$ for $m \prec n$.

So t_0 occurs only within S in t if in the abstract syntax tree, every path from the root t to an occurrence of t_0 as a subterm of t traverses some $t_1 \in S$ before reaching t_0.[2] If it occurs outside S, this means that $t_0 \sqsubseteq t$ and there is a path from the root to an occurrence of t_0 as a subterm of t that traverses no $t_1 \in S$.

Example 6 (Needham-Schroeder Occurrences). N_b occurs only within the singleton set $S_r = \{\{\!|N_a \hat{} N_b|\!\}_{\mathsf{pubk}(A)}\}$ in the term $\{\!|N_a \hat{} N_b|\!\}_{\mathsf{pubk}(A)}$. However, N_b occurs outside S_r in the term $\{\!|N_b|\!\}_{\mathsf{pubk}(B)}$, so N_b exits S_r passing from $\{\!|N_a \hat{} N_b|\!\}_{\mathsf{pubk}(A)}$ to $\{\!|N_b|\!\}_{\mathsf{pubk}(B)}$.

3.2 The Tests in Bundles

We say that a is *protected* in $\mathcal{B}$ iff $\mathsf{msg}(n) \neq a$ for all $n \in \mathcal{B}$. By the definitions of the penetrator strands for encryption and decryption (Definition 3), if the adversary uses K for encryption or decryption anywhere in $\mathcal{B}$, then K is not protected in $\mathcal{B}$. Thus, the adversary cannot create any encrypted term with a protected key K. If K^{-1} is protected, it cannot decrypt any term encrypted with K.

We say that a is *protected up to* m in $\mathcal{B}$, written $a \in \mathsf{Prot}_m(\mathcal{B})$, iff, for all $n \in \mathcal{B}$, if $\mathsf{msg}(n) = a$ then $m \prec_{\mathcal{B}} n$. If a key is protected up to a negative node m, then the adversary cannot use that key to prepare the term received on m.

Proposition 2 (Outgoing Authentication Test). *Suppose an atom a originates uniquely at a regular node n_0 in bundle $\mathcal{B}$, and suppose*

$$S \subseteq \{\{\!|t|\!\}_K : K^{-1} \in \mathsf{Prot}_{n_1}(\mathcal{B})\}.$$

If, for some $n_1 \in \mathcal{B}$, a exits S passing from $\mathsf{msg}(n_0)$ to $\mathsf{msg}(n_1)$, then a exits from S at *some positive* regular *$m_1 \preceq_{\mathcal{B}} n_1$. If n_0 and m_1 lie on different strands, then for some negative $m_0 \in \mathcal{B}$ with $a \sqsubseteq \mathsf{msg}(m_0)$,*

$$n_0 \prec_{\mathcal{B}} m_0 \Rightarrow^+ m_1 \preceq_{\mathcal{B}} n_1.$$

In the outgoing test, we call $m_0 \Rightarrow^+ m_1$ an *outgoing transforming edge* for a, S. It transforms the occurrence of a, causing a to exit S. We call (n_0, n_1) an *outgoing test pair for* a, S when a originates uniquely at n_0 and a exits S passing from $\mathsf{msg}(n_0)$ to $\mathsf{msg}(n_1)$. We also sometimes call m_1 an *outgoing transforming node* and n_1 an *outgoing test node.*

[2] In our terminology (Section 2), the K in $\{\!|t|\!\}_K$ is not an occurrence as a subterm.

Example 7. In the Needham-Schroeder protocol, with responder role s_r, the nodes $(s_r \downarrow 2), (s_r \downarrow 3)$ form an outgoing test pair for N_b, S_r, where S_r is as given in Example 6. If the initiator role is s_i, then the edge $s_i \downarrow 2 \Rightarrow s_i \downarrow 3$ is a outgoing transforming edge for N_b, S_r.

Also, the nodes $(s_i \downarrow 1), (s_i \downarrow 2)$ form an outgoing test pair for N_a, S_i, where S_i is the singleton set $\{\{\!|N_a \hat{} A|\!\}_{\mathsf{pubk}(C)}\}$. Letting $s'_r = s_r \cdot [B \mapsto C]$, then $s'_r \downarrow 1 \Rightarrow s'_r \downarrow 2$ forms an outgoing transforming edge for N_a, S_i.

Proposition 3 (Incoming Authentication Test). *Let $t = \{\!|t_0|\!\}_K$ with $K \in \mathsf{Prot}_{n_1}(\mathcal{B})$, and let $S \subseteq \{\{\!|t'|\!\}_{K_0} : K_0^{-1} \in \mathsf{Prot}_{n_1}(\mathcal{B})\}$. If t occurs outside S in any $n_1 \in \mathcal{B}$, then t exits S at some positive regular $m_1 \preceq_{\mathcal{B}} n_1$.*

We call m_1 an *incoming transforming node* for t, S, and n_1 an *incoming test node* for t, S. In our experience with existing protocols, Prop. 3 is always used with $S = \emptyset$, i.e. t does not occur at all before m_1. However, one can invent protocols requiring non-empty S, and completeness requires the stronger form.

4 Preskeletons, Skeletons, and Homomorphisms

The notion of a skeleton is intended to extract parts of the regular behavior of bundles, so that we can focus our inferences on what regular behavior must also be present.

4.1 Skeletons

A preskeleton is potentially the regular (non-penetrator) part of a bundle or of some portion of a bundle, and skeletons are the subset that are well-behaved, in that atoms intended to originate uniquely do so.

A preskeleton consists of nodes annotated with additional information, indicating order relations among the nodes, uniquely originating atoms, and non-originating atoms. We say that an atom a *occurs* in a set nodes of nodes if for some $n \in \mathsf{nodes}$, $a \sqsubseteq \mathsf{msg}(n)$. A key K is *used* in nodes if for some $n \in \mathsf{nodes}$, $\{\!|t|\!\}_K \sqsubseteq \mathsf{msg}(n)$. We say that a key K is *mentioned in* nodes if K or K^{-1} either occurs or is used in nodes. For a non-key a, a is mentioned if it occurs.

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

1. nodes is a finite set of regular nodes; $n_1 \in \mathsf{nodes}$ and $n_0 \Rightarrow^+ n_1$ implies $n_0 \in \mathsf{nodes}$;
2. $\preceq$ is a partial ordering on nodes such that $n_0 \Rightarrow^+ n_1$ implies $n_0 \preceq n_1$;
3. non is a set of keys, and for all $K \in \mathsf{non}$, either K or K^{-1} is used in nodes;

3′. for all $K \in \mathsf{non}$, K does not occur in nodes;

4. unique is a set of atoms, and for all $a \in \mathsf{unique}$, a occurs in nodes.

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

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

We select components of a preskeleton using subscripts, so, in $\mathbb{A} = (N, R, S, S')$, $\preceq_{\mathbb{A}}$ means R and $\mathsf{unique}_{\mathbb{A}}$ means S'. $\mathbb{A}$ need not contain all of the nodes of a strand, just some initial subsequence. We write $n \in \mathbb{A}$ to mean $n \in \mathsf{nodes}_{\mathbb{A}}$, 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 largest i with $s \downarrow i \in \mathbb{A}$. By Clauses 3, 4, $\mathsf{unique}_{\mathbb{A}} \cap \mathsf{non}_{\mathbb{A}} = \emptyset$.

Example 8. $\mathbb{A}_{ns}$, shown in Fig 1, is a skeleton with $\mathsf{non} = \{\mathsf{privk}(A)\}$, $\mathsf{unique} = \{N_b\}$. Its ordering is generated from the double arrows $\Rightarrow$, single arrows $\rightarrow$, and precedence signs. $\mathbb{A}_b$, containing only the responder strand s_r on the right side of Fig 1, is also a skeleton (equipped with $\mathsf{non} = \{\mathsf{privk}(A)\}$, $\mathsf{unique} = \{N_b\}$). However, if we adjoin a copy $s'_r = s_r \cdot [B \mapsto C]$ to $\mathbb{A}_{ns}$, then the result is not a skeleton, but only a preskeleton $\mathbb{A}_{pre}$. N_b originates both at $s_r \downarrow 2$ and at $s'_r \downarrow 2$. If instead we adjoin $s''_r = s_r \cdot [B \mapsto C, N_b \mapsto N'_b]$, we obtain a skeleton $\mathbb{A}'_{pre}$.

The skeletons for a protocol $\langle \Pi, \mathsf{strand\_non}, \mathsf{strand\_unique} \rangle$ are defined like the bundles for that protocol.

Definition 8. $\mathbb{A}$ is a *preskeleton for* protocol $\langle \Pi, \mathsf{strand\_non}, \mathsf{strand\_unique} \rangle$ iff for every $n \in \mathsf{nodes}_{\mathbb{A}}$ with $n = s \downarrow i$, (1) $s \in \Sigma_\Pi$; (2) if $s = r \cdot \alpha$ and $a \in \mathsf{strand\_non}_r \cdot \alpha$, then a does not occur in $\mathbb{A}$; and (3) if $s = r \cdot \alpha$ and $a \in \mathsf{strand\_unique}_r \cdot \alpha$, then $a \in \mathsf{unique}_{\mathbb{A}}$. $\mathbb{A}$ is a *skeleton for* a protocol if $\mathbb{A}$ is a skeleton, and $\mathbb{A}$ is a preskeleton for that protocol.

4.2 Skeletons and Bundles

Bundles correspond to certain skeletons:

Definition 9. Bundle $\mathcal{B}$ *realizes* skeleton $\mathbb{A}$ if:

1. The nodes of $\mathbb{A}$ are the regular nodes $n \in \mathcal{B}$.
2. $n \preceq_{\mathbb{A}} n'$ just in case $n, n' \in \mathsf{nodes}_{\mathbb{A}}$ and $n \preceq_{\mathcal{B}} n'$.
3. $K \in \mathsf{non}_{\mathbb{A}}$ iff case K or K^{-1} is used in $\mathsf{nodes}_{\mathbb{A}}$ but K occurs nowhere in $\mathcal{B}$.
4. $a \in \mathsf{unique}_{\mathbb{A}}$ iff a originates uniquely in $\mathcal{B}$.

The *skeleton* of $\mathcal{B}$ is the skeleton that it realizes. The skeleton of $\mathcal{B}$, written $\mathsf{skeleton}(\mathcal{B})$, is uniquely determined. $\mathbb{A}$ is *realized* if some $\mathcal{B}$ realizes it.

By condition (4), $\mathcal{B}$ does not realize $\mathbb{A}$ if $\mathbb{A}$ is a preskeleton but not a skeleton. Given a skeleton $\mathbb{A}$, methods derived from [6] determine whether $\mathbb{A}$ is realized. Skeleton $\mathbb{A}_{ns}$ from Example 8 is realized, but N_b is not.

Definition 10. A term t is *derivable before* n in $\mathbb{A}$ if there is a penetrator web G with $t \in R_G$ such that:

1. $S_G \subseteq \{\mathsf{msg}(m) \colon m \text{ positive and } m \preceq_{\mathbb{A}} n\}$;
2. If $K \in \mathsf{non}_{\mathbb{A}}$, K does not originate in G_n; and
3. If $a \in \mathsf{unique}_{\mathbb{A}}$ and a originates in $\mathbb{A}$, then a does not originate in G_n.

Proposition 4. *A skeleton $\mathbb{A}$ is realized iff, for every negative $n \in \mathbb{A}$, $\mathsf{msg}(n)$ is derivable before n in $\mathbb{A}$.*

4.3 Homomorphisms

When $\mathbb{A}$ is a preskeleton, we may apply a substitution α to it, subject to the same condition as in Prop. 1. Namely, suppose α is a replacement, and suppose that for each regular strand $s = r \cdot \beta$ such that s has nodes in $\mathbb{A}$, and for each atom $b \in u_r \cdot \beta$,

$$(\mathcal{O}(s, b)) \cdot \alpha = \mathcal{O}(s \cdot \alpha, b \cdot \alpha).$$

Then $\mathbb{A} \cdot \alpha$ is a well defined object. However, it is not a preskeleton when $x \cdot \alpha = y \cdot \alpha$ where $x \in \mathsf{non}_{\mathbb{A}}$ while y occurs in $\mathbb{A}$. In this case, no further identifications can restore the preskeleton property. So we are interested only in replacements with the property that $x \cdot \alpha = y \cdot \alpha$ and $x \in \mathsf{non}_{\mathbb{A}}$ implies y does not occur in $\mathbb{A}$. On this condition, $\mathbb{A} \cdot \alpha$ is a preskeleton.

However, $\mathbb{A}$ may be a skeleton, while objects built from it are preskeletons but not skeletons. In a preskeleton, we can sometimes, though, restore the skeleton unique origination property ($4'$) by a mapping ϕ that carries the two points of origination to a common node. This will be possible only if the terms on them are the same, and likewise for the other nodes in $\mathbb{A}$ on the same strands. We regard ϕ, α as an information-preserving, or more specifically information-increasing, map. It has added the information that a_1, a_2, which could have been distinct, are in fact the same, and thus the nodes n_1, n_2, which could have been distinct, must also be identified.

Example 9. $\mathbb{A}'_{pre}$ is a skeleton, but the result of applying the replacement $[N'_b \mapsto N_b]$ yields the preskeleton $\mathbb{A}_{pre}$ which is not a skeleton. If the map $\phi\colon \mathsf{nodes}_{\mathbb{A}_{pre}} \mapsto \mathsf{nodes}_{\mathbb{A}_{ns}}$ maps the successive nodes of the strand s'_r to the nodes of the strand s_r, then it will identify $s'_r \downarrow 2$ with $s_r \downarrow 2$, and thus restore the unique point of origination for N_b.

Definition 11. Let $\mathbb{A}_0, \mathbb{A}_1$ be preskeletons, α a replacement, $\phi\colon \mathsf{nodes}_{\mathbb{A}_0} \to \mathsf{nodes}_{\mathbb{A}_1}$. $H = [\phi, \alpha]$ is a *homomorphism* if

1a. For all $n \in \mathbb{A}_0$, $\mathsf{msg}(\phi(n)) = \mathsf{msg}(n) \cdot \alpha$, with the same direction;
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. $\mathsf{non}_{\mathbb{A}_0} \cdot \alpha \subseteq \mathsf{non}_{\mathbb{A}_1}$;
4. $\mathsf{unique}_{\mathbb{A}_0} \cdot \alpha \subseteq \mathsf{unique}_{\mathbb{A}_1}$; and $\phi(\mathcal{O}(s, a)) = \mathcal{O}(s', a \cdot \alpha)$ whenever $a \in \mathsf{unique}_{\mathbb{A}_0}$, $\mathcal{O}(s, a) \in \mathbb{A}_0$, and $\phi(s \downarrow j) = s' \downarrow j$.

We write $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ when H is a homomorphism from $\mathbb{A}_0$ to $\mathbb{A}_1$. When $a \cdot \alpha = a \cdot \alpha'$ for every a that occurs or is used for encryption in $\mathsf{dom}(\phi)$, then $[\phi, \alpha] = [\phi, \alpha']$; i.e., $[\phi, \alpha]$ is the equivalence class of pairs under this relation.

The condition for $[\phi, \alpha] = [\phi, \alpha']$ implies that the action of α on atoms not mentioned in the $\mathbb{A}_0$ is irrelevant. The condition on $\mathcal{O}$ in Clause 4 avoids the degeneracy in which a point of origination is destroyed for some atom $a \in \mathsf{unique}_{\mathbb{A}_0}$. We stipulate that such degenerate maps are not homomorphisms. For instance,

a replacement α that sends both N_a and N_b to the same value would not furnish homomorphisms on $\mathbb{A}_{ns}$. A responder, expecting to choose a fresh nonce, inadvertently selecting the same nonce N_a he has just received, would be an event of negligible probability. Thus, we may discard this degenerate set. Some homomorphisms are given in Example 2.

A homomorphism $I = [\phi, \alpha] \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ is an *isomorphism* iff ϕ is a bijection and there is an injective α' such that $[\phi, \alpha] = [\phi, \alpha']$. Two homomorphisms H_1, H_2 are isomorphic if they differ by an isomorphism I; i.e. $H_1 = I \circ H_2$.

When transforming a preskeleton $\mathbb{A}$ into a skeleton, one identifies nodes n, n' if some $a \in \mathsf{unique}_{\mathbb{A}}$ originates on both; to do so, one may need to unify additional atoms that appear in both $\mathsf{msg}(n), \mathsf{msg}(n')$. This process could cascade. However, when success is possible, and the cascading produces no incompatible constraints, there is a canonical (universal) way to succeed:

Proposition 5. *Suppose $H_0 \colon \mathbb{A} \mapsto \mathbb{A}'$ with $\mathbb{A}$ a preskeleton and $\mathbb{A}'$ a skeleton.*

There exists a homomorphism $G_{\mathbb{A}}$ and a skeleton $\mathbb{A}_0$ such that $G_{\mathbb{A}} \colon \mathbb{A} \mapsto \mathbb{A}_0$ and, for every skeleton $\mathbb{A}_1$ and every homomorphism $H_1 \colon \mathbb{A} \mapsto \mathbb{A}_1$, for some H, $H_1 = H \circ G_{\mathbb{A}}$. $G_{\mathbb{A}}$ and $\mathbb{A}_0$ are unique to within isomorphism.

Definition 12. The *hull* of $\mathbb{A}$, written $\mathsf{hull}(\mathbb{A})$, is the universal map $G_{\mathbb{A}}$ given in Prop. 5, when it exists. We write $\mathsf{hull}_{\alpha}(\cdot)$ for the partial map that carries any skeleton $\mathbb{A}$ to $\mathsf{hull}(\mathbb{A} \cdot \alpha)$.

We sometimes use the word *hull* to refer also to the target $\mathbb{A}_0$ of $G_{\mathbb{A}}$.

We say that a skeleton $\mathbb{A}_0$ is *live* if for some $H, \mathbb{A}_1$, $H \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ and $\mathbb{A}_1$ is realized. Otherwise, it is *dead*. There are two basic facts about dead skeletons:

Proposition 6 (Dead Skeletons). *(1) If $a \in \mathsf{non}_{\mathbb{A}}$ and $(\mathsf{Lsn}[a]) \downarrow 1 \in \mathbb{A}$, then $\mathbb{A}$ is dead. (2) If $\mathbb{A}$ is dead and $H \colon \mathbb{A} \mapsto \mathbb{A}'$, then $\mathbb{A}'$ is dead.*

4.4 Shapes

Shapes are minimal realizable skeletons, or more precisely, minimal homomorphisms with realizable targets.

Definition 13 (Shape). $[\phi, \alpha] \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ is *nodewise injective* if ϕ is an injective function on the nodes of $\mathbb{A}_0$.

A homomorphism H_0 is *nodewise less than or equal to* H_1, written $H_0 \leq_n H_1$, if for some nodewise injective J, $J \circ H_0 = H_1$. H_0 is *nodewise minimal* in a set S if $H_0 \in S$ and for all $H_1 \in S$, $H_1 \leq_n H_0$ implies H_1 is isomorphic to H_0.

$H \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ is a *shape for* $\mathbb{A}_0$ if H is nodewise minimal among the set of homomorphisms $H' \colon \mathbb{A}_0 \mapsto \mathbb{A}_1'$ where $\mathbb{A}_1'$ is realized.

The composition of two nodewise injective homomorphisms is nodewise injective, and a nodewise injective $H \colon \mathbb{A} \mapsto \mathbb{A}$ is an isomorphism. Thus, H_0, H_1 are isomorphic if each is nodewise less than or equal to the other. Hence, the relation $\leq_n$ is a partial order on homomorphisms, to within isomorphism.

When we say that $\mathbb{A}_1$ is a shape, we mean that it is the target of some shape $H \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$, where a particular $\mathbb{A}_0$ is understood from the context.

Proposition 7. *Let $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$. The set $\mathcal{S} = \{H'\colon H' \leq_n H\}$ is finite (up to isomorphism). If $\mathbb{A}_1$ is realized, then at least one $H' \in \mathcal{S}$ is a shape for $\mathbb{A}_0$.*

Example 10. The process described in this proof, applied to the embedding $H_{nsi}\colon \mathbb{A}_b \mapsto \mathbb{A}_{nsi}$ (see Example 2), discovers that the multiple occurrences of $\mathsf{pubk}(B)$ can be partitioned into those on the responder strand and those on the initiator strand. These can be distinguished, preserving being realized. Applied to the embedding of $\mathbb{A}_{b_2}$ (containing the first two responder node, see Example 1) into $\mathbb{A}_{nsi}$, it discards all the nodes outside $\mathbb{A}_{b_2}$, since the latter is already realized.

5 The Tests in Skeletons

To adapt the authentication tests of Section 3 to skeletons and homomorphisms, there are essentially two steps. First, we must "pull back" from bundles or realized skeletons to the skeletons that reach them via homomorphisms. Second, we can no longer read off the safe atoms from $\mathsf{Prot}(\mathcal{B})$. We have only partial information about which atoms will turn out to be safe or compromised. Thus, we speculatively consider both possibilities, i.e. both the possibility that a key will turn out to be compromised, and also the possibility that the transformed nodes need to be explained with a transforming edge.

Definition 14 (Augmentations, Contractions). 1. An *augmentation* is an inclusion $[\mathsf{id}, \mathsf{id}]\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ such that:
   (a) $\mathsf{nodes}_{\mathbb{A}_1} \setminus \mathsf{nodes}_{\mathbb{A}_0} = \{s \downarrow j\colon j \leq i\}$ for some $s = r \cdot \alpha$;
   (b) $\preceq_{\mathbb{A}_1}$ is the transitive closure of (i) $\preceq_{\mathbb{A}_0}$; (ii) the strand ordering of s up to i; (iii) pairs (n, m) or (n, m) with $n \in \mathsf{nodes}_{\mathbb{A}_0}$, $m = s \downarrow j$, and $j \leq i$; and (iv) the pair (n_a, m_a), when a originates on a node $n_a \in \mathbb{A}_0$ and a is mentioned in $m_a = s \downarrow j$, for any $a \in \mathsf{unique}_{\mathbb{A}_1}$.
   (c) $\mathsf{non}_{\mathbb{A}_1} = \mathsf{non}_{\mathbb{A}_0} \cup (\mathsf{strand\_non}_r \cdot \alpha)$; and
   (d) $\mathsf{unique}_{\mathbb{A}_1} = \mathsf{unique}_{\mathbb{A}_0} \cup (\mathsf{strand\_unique}_r \cdot \alpha)$.
2. An augmentation $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ is an *outgoing augmentation* if there exists an outgoing test edge $n_0, n_1 \in \mathbb{A}_0$ with no outgoing transforming edge in $\mathbb{A}_0$, and $s \downarrow 1 \Rightarrow^* m_0 \Rightarrow^+ s \downarrow i$, where $m_0 \Rightarrow^+ s \downarrow i$ is the earliest transforming edge for this test on s. The additional pairs in the ordering (clause 1b(iii)) are the pairs (n_0, m_0) and $((s \downarrow i), n_1)$.
3. It is an *incoming augmentation* if it adds an incoming transforming edge for an incoming test node in $\mathbb{A}_0$. The pair (m_1, n_1) in the notation of Prop. 3 is the additional pair in the ordering.
4. It is a *listener augmentation for a* if it adds a listener strand $\mathsf{Lsn}[a]$, with no pairs added to the ordering.
5. A replacement α is a *contraction* for $\mathbb{A}$ if there are two distinct atoms a, b mentioned in $\mathbb{A}$ such that $a \cdot \alpha = b \cdot \alpha$. We write $\mathsf{hull}_\alpha(\mathbb{A})$ for the canonical homomorphism from $\mathbb{A}$ to $\mathsf{hull}(\mathbb{A} \cdot \alpha)$, when the latter is defined. (See Prop. 5.)

Example 11. The embeddings H_{ns}, H_{nsi} (Example 2) are outgoing augmentations; the test edge lies between the second and third nodes of the responder strand. H_{ns} is more general, as H_{nsi} factors through it.

We use a listener strand $\mathsf{Lsn}[K]$, having the form $\stackrel{K}{\rightarrow} \bullet$ to mark a key K as a target for compromise. $\mathsf{Lsn}[K]$ records a commitment, the commitment to somehow compromise the value K before reaching a realized skeleton, if a transforming edge has not been chosen. The listener strand thus tests compromise for K. If K cannot be compromised, the skeleton containing the listener strand will be dead, and no homomorphism leads from it to a realized skeleton. Listener strands, lacking transmission nodes, never precede anything else; they are always maximal in $\preceq_{\mathbb{A}}$.

Since in a realized skeleton listener strands may be freely omitted, or freely added as long as the skeleton remains realized, we regard realized skeletons as *similar* if they differ only in what listener strands they contain. We write $\mathbb{A}_1 \sim_{\mathsf{L}} \mathbb{A}_2$ for skeletons that are similar in this sense. Shapes, being minimal, contain no listener strands; a homomorphism that simply embeds $\mathbb{A}_1$ into a $\mathbb{A}_2$ having more listener strands is nodewise injective.

We write $H_1 \sim_{\mathsf{L}} H_2$ if by adding listener strands we can equalize the homomorphisms H_1, H_2. That is, $H_1 \sim_{\mathsf{L}} H_2$ iff each H_i (for $i = 1, 2$) is of the form $H_i \colon \mathbb{A} \mapsto \mathbb{A}_i$, and there are embeddings $E_i \colon \mathbb{A}_i \mapsto \mathbb{A}'$ such that $\mathbb{A}_1 \sim_{\mathsf{L}} \mathbb{A}' \sim_{\mathsf{L}} \mathbb{A}_2$ and $E_1 \circ H_1 = E_2 \circ H_2$.

The search-oriented version of Prop. 2 states that when a skeleton $\mathbb{A}_0$ with an unsolved outgoing transformed pair leads to a realized skeleton $\mathbb{A}_1$, we can reach it starting with one of three kinds of steps: (1) a contraction, (2) an outgoing augmentation, or (3) adding a listener strand witnessing that one of the relevant keys is in fact *not* properly protected by the time we reach $\mathbb{A}_1$.

Theorem 1 (Outgoing Augmentation). *Let $H \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$, where $\mathbb{A}_1$ is realized. Let $n_0, n_1 \in \mathbb{A}_0$ be an outgoing test pair for a, S, for which $\mathbb{A}_0$ contains no transforming edge. Then there exist H', H'' such that either:*

1. *$H = H'' \circ H'$, and $H' = \mathsf{hull}_{\alpha}(\mathbb{A}_0)$ for some contraction α; or*
2. *$H = H'' \circ H'$, and H' is some outgoing augmentation for a, S; or*
3. *$H \sim_{\mathsf{L}} H'' \circ H'$, and H' is a listener augmentation $H' \colon \mathbb{A}_0 \mapsto \mathbb{A}_0'$ adding $\mathsf{Lsn}[K^{-1}]$, for some $K \in \mathsf{used}(S)$.*

Proof. Assuming $H = [\phi, \alpha] \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ with $\mathbb{A}_1$ realized, say with $\mathsf{skeleton}(\mathcal{B}) = \mathbb{A}_1$, we have the following possibilities. If α contracts any atoms, then we may factor H into a contraction followed by some remainder H'' (clause 1).

If α does not contract any atoms, then $(\phi(n_0), \phi(n_1))$ is an outgoing test pair for $a \cdot \alpha, S \cdot \alpha$. There are now two cases. First, suppose $\mathsf{used}(S) \cdot \alpha \subseteq \mathsf{Prot}_{\phi(n_1)}(\mathcal{B})$. Then we may apply Prop. 2 to infer that $\mathcal{B}$ and thus also $\mathbb{A}_1$ contains an outgoing transforming edge $m_0 \Rightarrow^+ m_1$ for $a \cdot \alpha, S \cdot \alpha$. Since α is injective on atoms mentioned in $\mathbb{A}_0$, we may augment $\mathbb{A}_0$ with an edge $m_0' \Rightarrow^+ m_1'$ such that $\mathsf{msg}(m_0') \cdot \alpha = \mathsf{msg}(m_0)$ and $\mathsf{msg}(m_1') \cdot \alpha = \mathsf{msg}(m_1)$.

Second, if there is some $K \in \mathsf{used}(S)$ such that $K^{-1} \cdot \alpha \notin \mathsf{Prot}_{\phi(n_1)}\mathcal{B}$, then there is $\mathbb{A}_1' \sim_{\mathsf{L}} \mathbb{A}_1$ such that $\mathbb{A}_1'$ contains $\mathsf{Lsn}[K^{-1} \cdot \alpha]$, and $\phi(n_1) \not\preceq (\mathsf{Lsn}[K^{-1} \cdot \alpha]) \downarrow 1$. Hence, clause 3 is satisfied.

Incoming augmentations are similar to outgoing ones, except that the key used for encryption in the test node is also relevant. The proof is similar.

Theorem 2 (Incoming Augmentation). *Let $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$, where $\mathbb{A}_1$ is realized. Let $n_1 \in \mathbb{A}_0$ be an incoming test node for t, S with $t = \{|t_0|\}_K$. If there is no incoming transforming node for t, S in $\mathbb{A}_0$, then there exist H', H'' such that either:*

1. *$H = H'' \circ H'$, and $H' = \mathsf{hull}_\alpha(\mathbb{A}_0)$ for some contraction α; or*
2. *$H = H'' \circ H'$, for H' an incoming augmentation emitting $\{|t_0|\}_K$ occurring outside S; or*
3. *$H \sim_{\mathsf{L}} H'' \circ H'$, for H' a listener augmentation $H'\colon \mathbb{A}_0 \mapsto \mathbb{A}'_0$ adding K or some K_0^{-1}, for $K_0 \in \mathsf{used}(S)$.*

Evidently, Thms. 1–2 are useful for constructing shapes. They say in effect that any shape $H_s\colon \mathbb{A}_0 \mapsto \mathbb{A}_s$ may be factored into a composition $H'' \circ H'$, where H' is dictated by Thm. 1 or Thm. 2, and H'' can be determined by repeating this process. Since $\mathbb{A}_s$ is a finite structure, presumably this process must terminate in each case, although one cannot predict in advance how many steps might be needed [4]. In related work, we have in fact proved that every shape may in fact be obtained through this process [3, ext.vers.]

References

1. Martín Abadi and Bruno Blanchet. Analyzing security protocols with secrecy types and logic programs. *Journal of the ACM*, 52(1):102–146, January 2005.
2. Roberto M. Amadio and Denis Lugiez. On the reachability problem in cryptographic protocols. In *Concur*, number 1877 in LNCS, pages 380–394, 2000.
3. Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer. Searching for shapes in cryptographic protocols. In *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, number 4424 in LNCS, pages 523–538. Springer, March 2007. Extended version at URL:`http://eprint.iacr.org/2006/435`.
4. 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.
5. Andrew D. Gordon and Alan Jeffrey. Types and effects for asymmetric cryptographic protocols. *Journal of Computer Security*, 12(3/4):435–484, 2003.
6. Joshua D. Guttman and F. Javier Thayer. Authentication tests and the structure of bundles. *Theoretical Computer Science*, 283(2):333–380, June 2002. Conference version appeared in *IEEE Symposium on Security and Privacy*, May 2000.
7. ITU. Message sequence chart (MSC). Recommendation Z.120, 1999.
8. Leslie Lamport. Time, clocks and the ordering of events in a distributed system. *CACM*, 21(7):558–565, 1978.
9. 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.
10. Jonathan K. Millen and Vitaly Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *8th ACM Conference on Computer and Communications Security (CCS '01)*, pages 166–175. ACM, 2001.
11. Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12), 1978.