Searching for Shapes in Cryptographic Protocols (extended version)*

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

The MITRE Corporation

Abstract. We describe a method for enumerating all essentially different executions possible for a cryptographic protocol. We call them the *shapes* of the protocol. Naturally occurring protocols have only finitely many, indeed very few shapes. Authentication and secrecy properties are easy to determine from them, as are attacks and anomalies. CPSA, our Cryptographic Protocol Shape Analyzer, implements the method.

In searching for shapes, CPSA starts with some initial behavior, and discovers what shapes are compatible with it. Normally, the initial behavior is the point of view of one participant. The analysis reveals what the other principals must have done, given this participant's view.

The search is *complete*, i.e. every shape can in fact be found in a finite number of steps. The steps in question are applications of two *authentication tests*, fundamental patterns for protocol analysis and heuristics for protocol design. We have formulated the authentication tests in a new, stronger form, and proved completeness for a search algorithm based on them.

1 Introduction

The executions of cryptographic protocols frequently have very few essentially different forms, which we call *shapes*. By enumerating these shapes, we may ascertain whether they all satisfy a security condition such as an authentication or confidentiality property. We may also find other anomalies, which are not necessarily counterexamples to the security goals, such as involving unexpected participants, or involving more local runs than expected.

In this paper, we describe a complete method for enumerating the shapes of a protocol within a pure Dolev-Yao model [7]. If the protocol has only finitely many essentially different shapes, the enumeration will terminate. From the shapes, we can then read off the answers to secrecy and authentication questions and

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

observe other anomalies. Our software implementation of this method is called a Cryptographic Protocol Shapes Analyzer (CPSA).

We use the strand space theory [10]. 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 [13] or fragments of message sequence charts [12]. A skeleton is *realized* if it is nonfragmentary, i.e. it contains exactly the regular behavior of some execution. A realized skeleton is a *shape* if it is minimal in a sense we will make precise (Definition 14). We *search* for shapes using the authentication tests [10] to find new strands to add when a skeleton is not large enough to be realized.

The main technical result underlying CPSA is *completeness*, in the sense that—for any protocol—our authentication test search eventually discovers every shape for that protocol (Thm. 3). It cannot terminate for every protocol [8]. It does, however, terminate for reasonably inclusive classes [4, 19].

The type-and-effect system for spi calculus [9] is related to the authentication tests, but differs from our work in two ways. First, we do not use the syntactically-driven form of a type system, but instead a direct analysis of behaviors. Second, type-and-effect systems aim at a sound approximation, whereas our work provides actual counterexamples when a security goal is not met. Blanchet's ProVerif [1] is also based on a sound approximation, and may thus refuse to certify a protocol even though there are no counterexamples.

CPSA's search is related to the second version of Athena [18], which adopted the authentication tests from [10]. However, CPSA differs from Athena in several ways. First, it involves the regular behaviors alone; we never represent adversary activity within a shape. Second, the notion of shape defines a criterion for which possible executions should be considered, among the infinitely many executions (of unbounded size) of any protocol. Third, we introduce strong versions of the authentication tests, for which completeness is true.

The shapes describe protocol executions of all sizes; we do not follow the widely practiced *bounded* protocol analysis (e.g. [2, 15]).

Structure of this paper. Section 2 describes the idea of shapes, using the simplest example. Section 3 summarizes strand space terminology, with new definitions for *replacements* and *protocols*. Section 4 states the new version of the authentication test theorems as properties of bundles, i.e. complete protocol executions including adversary behavior. In Section 5 we define skeletons and homomorphisms, and in Section 6 we infer forms of the authentication tests that concern homomorphisms from skeletons to realized skeletons.

In Section 7 we illustrate the process using the Yahalom protocol. This analysis illustrates almost every aspect of the CPSA search method. In Section 8, we define the search's control structure. Section 9 establishes completeness.

Finally, the CPSA implementation is the subject of Section 10.

A shorter version of this work, focusing on the search but not its completeness, appears as [6].

2 The Idea of Shapes

In practice, protocols have remarkably few shapes. The Needham-Schoeder-Lowe [16, 14] 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.

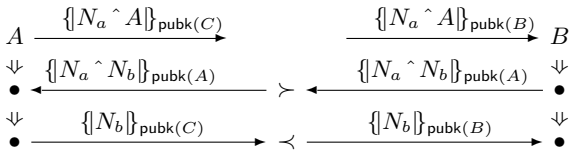

Fig. 1. Needham-Schroeder Shape for B ($\mathsf{privk}(A)$ uncompromised, N_b fresh)

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 3.2.)

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 3.1.)

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 3.2, 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 [14]; 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** $\mathbb{A}$ 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. 8.)

$\mathbb{A}$ 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. 10.)

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, 12.)

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

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

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.

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. 14.)

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. 8.)

3 Terms, Strands, and Bundles

In this section and Section 5 we give precise definitions, which include a number of fine points which seemed an unnecessary distraction in Section 2. In this section, the definitions of replacement and protocol (Defs. 1, 4) are new versus [10].

3.1 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 mapping principals to keys, or an injective *long term shared key of* function mapping pairs of principals to keys. These functions are not constructors, and their results are atoms. For definiteness, we include here functions $\mathsf{pubk}(a), \mathsf{ltk}(a)$ mapping principals to (respectively) their public keys and to a symmetric key shared on a long-term basis with a fixed server S. $\mathsf{pubk}(a)^{-1}$ 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 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 $\mathsf{tag}\,\hat{}\,t_0\,\hat{}\,t_1$. 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 $x \cdot \alpha = x$.

3.2 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: $\langle +K \rangle$

$\mathsf{C}_{g,h}$: $\langle -g,\ -h,\ +g\,\hat{}\,h \rangle$ — $\mathsf{S}_{g,h}$: $\langle -g\,\hat{}\,h,\ +g,\ +h \rangle$

$\mathsf{E}_{h,K}$: $\langle -K,\ -h,\ +\{|h|\}_K \rangle$ — $\mathsf{D}_{h,K}$: $\langle -K^{-1},\ -\{|h|\}_K,\ +h \rangle$.

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) $t_0 \sqsubseteq \mathsf{tag}\,\hat{}\,t_0\,\hat{}\,t_1$; (2) $t_1 \sqsubseteq \mathsf{tag}\,\hat{}\,t_0\,\hat{}\,t_1$; 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. 12, 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 $\langle -K \rangle$.

3.3 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. 9 and 15, 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 [16] 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 $(\mathcal{O}(s, a)) \cdot \alpha = \mathcal{O}(s \cdot \alpha, a \cdot \alpha)$.

4 Strengthened Authentication Tests in Bundles

To direct the process of searching for realized skeletons, we use the *authentication tests* [10] in a strengthened and simplified form.

4.1 "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 $\{\!|N_b|\!\}_{\mathsf{pubk}(B)}$.

4.2 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*

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

[2] In our terminology (Section 3), the K in $\{\!|t|\!\}_K$ is not an occurrence as a subterm.

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 $a \sqsubseteq \mathsf{msg}(m_0)$,*

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

Proof. Apply Prop. 1 to $T =$

$$\{m \colon m \preceq_{\mathcal{B}} n_1 \text{ and } a \text{ occurs outside } S \text{ in } \mathsf{msg}(m)\};$$

$n_1 \in T$, so T has $\preceq_{\mathcal{B}}$-minimal members m_1. Since keys K used in S have $K^{-1} \in \mathsf{Prot}_{n_1}(\mathcal{B})$, m_1 cannot lie on a decryption penetrator D-strand. By unique origination, a does not lie on a M-strand or K-strand. By the definitions of S and "occurs only within," m_1 does not lie on a S-, C-, or E-strand. Thus, m_1 lies on some $s \in \Sigma_\Pi$. If n_0 does not lie on s, then a does not originate on s, so $a \sqsubseteq \mathsf{msg}(m_0)$ for some negative m_0, with $m_0 \Rightarrow^+ m_1$. □

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.*

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 $S \subseteq \{\{\!|t'|\!\}_{K_0} \colon K_0^{-1} \in \mathsf{Prot}_{n_1}(\mathcal{B})\}$. If t occurs outside S in any $n_1 \in \mathcal{B}$, then t exits S at some positive regular $m_1 \preceq_{\mathcal{B}} n_1$.*

Proof. Apply Prop. 1 to the set $T =$

$$\{m \colon m \preceq_{\mathcal{B}} n_1 \text{ and } t \text{ occurs outside } S \text{ in } \mathsf{msg}(m)\};$$

$n_1 \in T$, so T has minimal members m_1. Since keys K_0 used in S have $K_0^{-1} \in \mathsf{Prot}_{n_1}(\mathcal{B})$, m_1 cannot lie on a decryption D-strand. Since $K \in \mathsf{Prot}_{n_1}(\mathcal{B})$, m_1 cannot lie on an encryption E-strand. The remaining penetrator strands are inapplicable by the definition of "occurs only within". □

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.

4.3 Penetrator Webs and Test Nodes

We can see that Props. 2–3 have some sort of completeness by considering the powers of the adversary. In essence, if any negative regular node is neither an outgoing test node nor an incoming test node, then the adversary can derive the term on it. Thus, only test nodes in this sense can provide authentication guarantees about the presence of regular activity. The rest could be the adversary's work.

To make this precise, we define penetrator webs, which characterize what the adversary can do with fixed inputs from the regular participants.

Definition 7 (Penetrator web, derivable). Let $G = \langle \mathcal{N}_G, (\rightarrow_G \cup \Rightarrow_G) \rangle$ be a finite acyclic subgraph of $\langle \mathcal{N}, (\rightarrow \cup \Rightarrow) \rangle$ such that $\mathcal{N}_G$ consists entirely of penetrator nodes. G is a *penetrator web* with support S_{spt} and result R if S_{spt} and R are sets of terms and moreover:

1. If $n_2 \in \mathcal{N}_G$ is negative, then either $\mathsf{msg}(n_2) \in S_{\mathrm{spt}}$ or there is a unique n_1 such that $n_1 \rightarrow_G n_2$.
2. If $n_2 \in \mathcal{N}_G$ and $n_1 \Rightarrow n_2$ then $n_1 \Rightarrow_G n_2$.
3. For each $t \in R$, either $t \in S_{\mathrm{spt}}$ or for some positive $n \in \mathcal{N}_G$, $\mathsf{msg}(n) = t$.

If A is a set of atoms, then term t_1 is *derivable from* S_{spt} *avoiding* A if there is a web G with support $S_G \subseteq S_{\mathrm{spt}}$ and $t_1 \in R_G$, where no atom in A originates on a penetrator strand in G.

If $n \in \mathcal{B}$ is a negative node, then $\mathcal{B}$ includes a penetrator web G with result $R_G = \{\mathsf{msg}(n)\}$. Its support $S_G = \{\mathsf{msg}(m) \colon m \text{ is positive regular and } m \prec_{\mathcal{B}} n\}$.

When S_{spt} is a set of terms, we say that t *exits* S_{enc} *passing from* S_{spt} *to* t_1 if for each $t_0 \in S_{\mathrm{spt}}$, t exits S_{enc} passing from t_0 to t_1. Def. 6 says that this means that t occurs only within the encryptions in S_{enc} in every $t_0 \in S_{\mathrm{spt}}$, and t occurs outside S_{enc} in t_1.

Suppose that $\mathcal{B}$ is a bundle, and U is the set of atoms that originate uniquely in $\mathcal{B}$, and on a regular node. In the following proposition, letting $A = U \cup \mathsf{Prot}_{n_1}(\mathcal{B})$, the first condition says that $t_1 \neq \mathsf{msg}(n_1)$ for any outgoing test node $n_1 \in \mathcal{B}$. The second condition says that $t_1 \neq \mathsf{msg}(n_1)$ for any incoming test node $n_1 \in \mathcal{B}$.

Proposition 4. *Let A be a set of atoms; let S_{spt} be a finite set of terms; and let t_1 be a term such that, for any $a \in A$, if $a \sqsubseteq t_1$, then $a \sqsubseteq t_0$ for some $t_0 \in S_{\mathrm{spt}}$. Suppose the following conditions hold:*

1. *for all $a \in A$ and all sets of encryptions S_{enc}, if a exits S_{enc} passing from S_{spt} to t_1, then there is some $\{\!|t|\!\}_{K_0} \in S_{\mathrm{enc}}$, such that ${K_0}^{-1}$ is derivable from S_{spt} avoiding A; and*
2. *for all encryptions $\{\!|t|\!\}_K$, and all sets of encryptions S_{enc}, if $\{\!|t|\!\}_K$ exits S_{enc} passing from S_{spt} to t_1, then either K is derivable from S_{spt} avoiding A, or else some ${K_0}^{-1}$ with $K_0 \in \mathsf{used}(S_{\mathrm{enc}})$ is derivable from S_{spt} avoiding A.*

Then term t_1 is derivable from S_{spt} avoiding A.

Proof. The proof is by structural induction on the pair (S_{spt}, t_1), i.e. the ordering under which $(S_{\mathrm{spt}}, t_1) \leq (S'_{\mathrm{spt}}, t'_1)$ iff $t_1 \sqsubseteq t'_1$, and for all $t \in S_{\mathrm{spt}}$, there is some $t' \in S'$ such that $t \sqsubseteq t'$.

Case $t_1 = a$**:** If $a \notin A$, then the one-node web originating a satisfies the conditions. Otherwise, $S^a = \{t \in S_{\mathrm{spt}} \colon a \sqsubseteq t\}$ is non-empty. If $a \in S^a$, then the empty web suffices. If some concatenation $t_0 \hat{} t'_0 \in S^a$, then apply the induction hypothesis to $S^a \setminus \{t_0 \hat{} t'_0\} \cup \{t_0\} \cup \{t'_0\}$. This asserts the existence of a penetrator web G_a deriving a. Obtain the desired web by prepending a separation S-strand above any occurrences of t_0 and t'_0 in G_a.
Otherwise, S^a consists entirely of encryptions, and a exits S^a passing from S_{spt} to a. By condition 1, there is some $\{\!|t|\!\}_{K_0} \in S^a$ with ${K_0}^{-1}$ derivable from S_{spt} avoiding A, using some web $G_{{K_0}^{-1}}$. Thus, applying the induction hypothesis to $(S^a \setminus \{\{\!|t|\!\}_{K_0}\}) \cup \{t\}$, we obtain a web G. We may prepend $G_{{K_0}^{-1}}$ and a decryption D-strand before G to obtain the required web.

Case $t_1 = t'_1 \hat{} t''_1$**:** Apply the induction hypothesis to t'_1 and t''_1, and append a concatenation C-strand after the resulting webs.

Case $t_1 = \{\!|t'_1|\!\}_K$**:** Suppose K is derivable from S_{spt} avoiding A, using some web G_K. Apply the induction hypothesis to t'_1, obtaining a web G. Append an encryption E-strand after G_K and G to derive $\{\!|t'_1|\!\}_K$.
Otherwise, by condition 2, some K_0^{-1} with $\{\!|t_0|\!\}_{K_0} \in S_{\mathrm{enc}}$ is derivable from S_{spt} avoiding A, using a web $G_{{K_0}^{-1}}$. Apply the induction hypothesis to $(S_{\mathrm{spt}} \setminus \{\{\!|t_0|\!\}_{K_0}\}) \cup \{t_0\}$, obtaining a web G. Prepend $G_{{K_0}^{-1}}$ and a decryption D-strand before G. □

5 Preskeletons, Skeletons, and Homomorphisms

The notion of penetrator web from Section 4.3 extracts parts of the adversary activity in a bundle, helping us focus on its structure. 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.

5.1 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 8. 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 9. $\mathbb{A}$ 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.

5.2 Skeletons and Bundles

Bundles correspond to certain skeletons:

Definition 10. 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 [10] determine whether $\mathbb{A}$ is realized. Skeleton $\mathbb{A}_{ns}$ from Example 8 is realized, but N_b is not.

Definition 11. 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 5. *A skeleton $\mathbb{A}$ is realized iff, for every negative $n \in \mathbb{A}$, $\mathsf{msg}(n)$ is derivable before n in $\mathbb{A}$.*

5.3 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 12. 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 6. *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 13. The *hull* of $\mathbb{A}$, written $\mathsf{hull}(\mathbb{A})$, is the universal map $G_{\mathbb{A}}$ given in Prop. 6, 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 7 (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.*

5.4 Shapes

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

Definition 14 (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 8. *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 $\mathbb{A}_0$.*

Proof. Letting $H = [\phi, \alpha]$, we generate $\mathcal{S}$ by making two choices. For each node $n \in (\mathbb{A}_1 \setminus \phi(\mathbb{A}_0))$, we consider whether to omit it and all nodes later than n on the same strand. Second, we also consider, whenever two strands contain the same parameter a, whether to assign these parameters different values.

We partition the strands containing a parameter a. A partition is permissible if it respects identifications already present in $\mathbb{A}_0$. By this we mean that s'_0, s'_1 are in the same partition class whenever there are strands s_0, s_1 with $\phi(s_0) = s'_0$ and $\phi(s_1) = s'_1$, and s_0, s_1 already agree on this parameter. We choose a partition for each parameter a.

$\mathcal{S}$ contains the homomorphisms determined by a choice of nodes to omit and a family of permissible partitions. We replace each occurrence of a either by a representative atom from $\alpha^{-1}(a)$ or by a new value not mentioned in $\mathbb{A}_0$. There are only finitely many ways to do this (to within renaming), so $\mathcal{S}$ is finite.

H is a member of $\mathcal{S}$, namely the one that omits no nodes and has a single partition class for each a. Thus, if $\mathbb{A}_1$ is realized, $\mathcal{S}$ has members with a realized target. Letting $\mathcal{S}' \subseteq \mathcal{S}$ be the set of $H' \in \mathcal{S}$ such that the target of H' is realized, $\mathcal{S}'$ is non-empty and finite; hence, $\mathcal{S}'$ has $\leq_n$-minimal members. □

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.

6 The Tests in Skeletons

To adapt the authentication tests of Section 4 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 15 (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. 6.)

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.

Protocols have only finitely many roles, hence only finitely many transforming edges that can solve a given outgoing test pair n_0, n_1. Each S uses only finitely many keys. Moreover, each $\mathbb{A}_0$ mentions only finitely many atoms. Thus, there are only finitely many (non-isomorphic) contractions.

Proposition 9. *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. There are finite families $\mathcal{F} = \{H'_j\}_{j \leq k}$ of contractions and augmentations as in Theorem 1, such that for every $H \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$, where $\mathbb{A}_1$ is realized, $H \sim_{\mathsf{L}} H'' \circ H'_j$ for some H'' and $H'_j \in \mathcal{F}$.*

For each unsolved $n_0, n_1 \in \mathbb{A}_0$ and a, S, there is a finite, most general set $\mathcal{F}$ of contractions and augmentations H solving it, meaning that any such H factors through some member of $\mathcal{F}$.

The outgoing augmentations in a most general family are obtained by unifying messages in the roles of Π with terms in S. One selects edges lying between a negative node in which a occurs only within $S \cdot \alpha$ and a positive node in which

a occurs outside $S \cdot \alpha$. We need a contraction α only if either (1) $n_0 \cdot \alpha, n_1 \cdot \alpha$ is no longer an outgoing transformed pair, or else (2) for some candidate outgoing augmentation, $n_0 \cdot \alpha, n_1 \cdot \alpha$ is the most general version of the test that it solves. This occurs when the protocol role mentions the same atom at several locations where different atoms are mentioned in n_0, n_1; α must then identify these atoms.

We call a most general $\mathcal{F}$ an *outgoing cohort* for n_0, n_1 and a, S.

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 $K_0 \in \mathsf{used}(S)$.*

Each incoming test determines a most general family of augmentations and contractions, akin to those in Prop. 9; we refer to them as *incoming cohorts*.

Evidently, Thms. 1–2 are useful for authentication results. Thm. 1 is also useful for checking secrecy for atoms a. Starting from $\mathbb{A}_0$, we add a listener $\mathsf{Lsn}[a]$ to $\mathbb{A}_0$. If a is not penetrator-derivable, then $\mathsf{Lsn}[a]$ is an outgoing test node. Incoming and outgoing augmentations may lead from $\mathbb{A}_0 \cup \mathsf{Lsn}[a]$ to a realized skeleton. Success guarantees that a can become compromised, while failure ensures its safety.

7 The Yahalom Protocol

Yahalom's protocol is quite compact, but subtle. The shape search for the responder illustrates almost every aspect of the CPSA search method.

7.1 Protocol Definition

The Yahalom protocol (Fig. 2 [17]) provides a session key K to principals sharing long-term symmetric keys with a key server. We let $\mathsf{ltk}(\cdot)$ map each principal A to its long term shared key $\mathsf{ltk}(A)$. We assume that all participants agree on the server, which does not also participate as a client.

The protocol contains three roles, the initiator, the responder, and the server. Each is described by one strand in Fig. 2, each parametrized by A, B, N_a, N_b, K. The parameters are atomic values, and the instances of each role are constructed by replacing them with other atomic values. The behavior Init of the initiator consists in transmitting $A \hat{} N_a$ followed by receiving some message of the form $\{|B \hat{} K \hat{} N_a \hat{} N_b|\}_{\mathsf{ltk}(A)}$ and finally transmitting $\{|N_b|\}_K$. The other roles are also self-explanatory. The key server is trusted to generate a fresh, uniquely originating session key K in each run, i.e. $\mathsf{strand\_unique}_{\mathsf{Serv}} = \{K\}$.

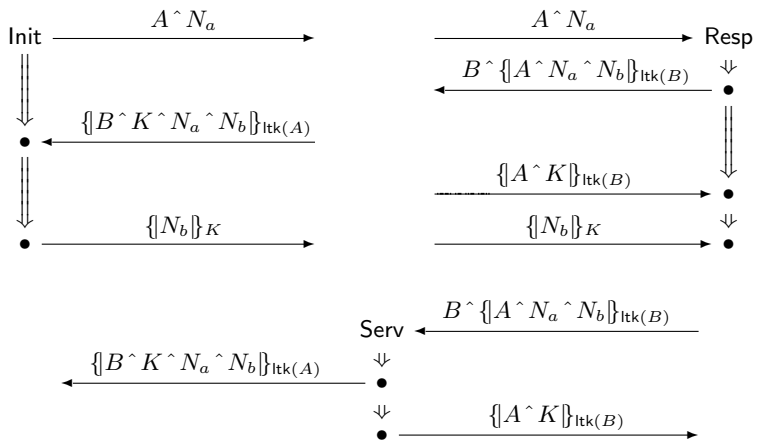

Fig. 2. Yahalom protocol (forwarding removed)

7.2 A Search: Shapes for the Responder

Suppose an execution contains a local run s_r of the responder's role as in the upper right column of Fig. 2. We assume the long term keys $\mathsf{ltk}(A), \mathsf{ltk}(B)$ are uncompromised, as no authentication can be achieved otherwise. Similarly, we assume the responder's nonce N_b to be fresh and unguessable.

So let the initial skeleton $\mathbb{A}_0$ consist of s_r, with $\mathsf{non}_{\mathbb{A}_0} = \{\mathsf{ltk}(A), \mathsf{ltk}(B)\}$ and $\mathsf{unique}_{\mathbb{A}_0} = \{N_b\}$. What skeletons are shapes for $\mathbb{A}_0$? Or more precisely, for what realized skeletons $\mathbb{A}$ is there a shape $H \colon \mathbb{A}_0 \mapsto \mathbb{A}$?

We will find only one possibility, the skeleton $\mathbb{A}_4$ (Fig. 5). Any realized $\mathbb{A}$ containing any responder strand s'_r—with uncompromised long-term keys and a fresh nonce—has a subskeleton $\mathbb{A}'$ containing s'_r, with $J \colon \mathbb{A}_4 \mapsto \mathbb{A}'$. J is both nodewise injective and surjective, i.e. an isomorphism on nodes, although it may identify atoms. In this sense, $\mathbb{A}'$, the portion of $\mathbb{A}$ containing s'_r, resembles $\mathbb{A}_4$.

Transforming the Nonce. B chooses a fresh nonce N_b in node n_0 (see Fig. 3), and transmits it within the encrypted unit $\{\!|A \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(B)}$. In B's node n_2, it is received outside that unit, in the form $\{\!|N_b|\!\}_K$. So n_0, n_2 is an outgoing test pair for N_b, S_1 where $S_1 =$

$$\{\{\!|A \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(B)}\} \cup \{\{\!|B \hat{} K' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)} \colon K' \text{ a key}\}.$$

1. The only outgoing transforming edges for N_b, S_1 lie on initiator strands. Unifying node 2 of the role with messages in S_1 shows that the parameters must be A, B, N_a, N_b, and some K'. We ask later whether $K' = K$.
2. Alternatively some decryption key may be compromised. Since $\mathsf{used}(S_1) = \{\mathsf{ltk}(A), \mathsf{ltk}(B)\}$ contains symmetric (self-inverse) keys, this means we consider adding $\mathsf{Lsn}[\mathsf{ltk}(A)]$ or $\mathsf{Lsn}[\mathsf{ltk}(B)]$.

No contraction is relevant. Thus, these three embeddings—adding to $\mathbb{A}_0$ either an initiator strand s_i, or a listener strand $\mathsf{Lsn}[\mathsf{ltk}(A)]$, or a listener $\mathsf{Lsn}[\mathsf{ltk}(B)]$—form an outgoing cohort. When adding s_i, we know that $n_0 \prec (s_i \downarrow 2) \Rightarrow (s_i \downarrow 3) \prec n_2$.

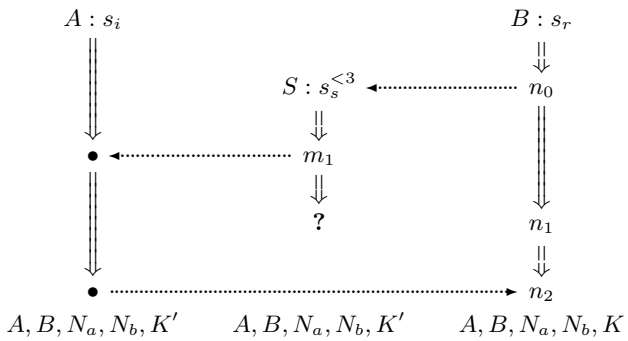

Fig. 3. Skeleton $\mathbb{A}_1$, with $\mathsf{non}_{\mathbb{A}_1} = \{\mathsf{ltk}(A), \mathsf{ltk}(B)\}$ and $\mathsf{unique}_{\mathbb{A}_1} = \{N_b, K'\}$.

Since $\mathsf{non}_{\mathbb{A}_0} = \{\mathsf{ltk}(A), \mathsf{ltk}(B)\}$, Prop. 7 says that $\mathbb{A}_0 \cup \mathsf{Lsn}[\mathsf{ltk}(A)]$ and $\mathbb{A}_0 \cup \mathsf{Lsn}[\mathsf{ltk}(B)]$ are unrealizable. No bundle $\mathcal{B}$ can ever contain a listener strand for a value that originates nowhere. Thus, the embeddings of Case 2 are dead. Thus, every homomorphism from $\mathbb{A}_0$ to a realized skeleton factors through the embedding $\mathbb{A}_0 \mapsto \mathbb{A}_0 \cup \{s_i\}$ of Case 1.

We again have an outgoing test edge between n_0 and $s_i \downarrow 2$, for N_b, S_2 where

$$S_2 = \{\{\!|A \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(B)}\} \cup \{\{\!|B \hat{} K'' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)} \colon K'' \neq K'\}.$$

N_b originates only at n_0, where it occurs only within S_2; however, in $\mathsf{msg}(s_i \downarrow 2)$, N_b occurs outside S_2 in the form $\{\!|B \hat{} K' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)}$.

3. The only outgoing transforming edges for N_b, S_2 lie on server strands s_s. Unifying node 1 of the role with messages in S_2 shows that the parameters must be A, B, N_a, N_b, and some K''. Since N_b must occur outside S_2 in $s_s \downarrow 2$, we have $K'' = K'$; so that the last parameter is K'. The last node $s_s \downarrow 3$ may not be included; we will write $s_s^{<3}$ for the initial segment of s_s.
4. Alternatively a decryption key in $\mathsf{used}(S_1) = \{\mathsf{ltk}(A), \mathsf{ltk}(B)\}$ may be compromised. However, the listener strands produce dead skeletons, as in Case 2.

Thus, any homomorphism from $\mathbb{A}_0$ to a realized skeleton must factor through the embedding $\mathbb{A}_0 \mapsto \mathbb{A}_0 \cup s_r \cup s_s^{<3}$. We call this skeleton $\mathbb{A}_1$, shown in Fig 3, which also shows how the ordering relation extends. Since the server always provides a fresh session key, we also have $K' \in \mathsf{unique}_{\mathbb{A}_1}$.

Does $K' = K$? The server generated K' on strand s_s and delivers it to A on $s_i \downarrow 2$. B receives K on n_1, and on n_2 finds K also used to encrypt the nonce N_b. Must the keys K', K be the same, or could they be distinct?

Nodes n_0, n_2 form an outgoing test pair for N_b and the set

$$S_3 = \{\, \{\!|A \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(B)}, \{\!|B \hat{} K' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)}, \{\!|N_b|\!\}_{K'} \,\}.$$

The resulting outgoing cohort consists of Cases 5–7:

5. Under the contraction β that maps $K' \mapsto K$ and is elsewhere the identity, no new edge is needed, as $\{\!|N_b|\!\}_{K'} \cdot \beta = \{\!|N_b|\!\}_K \cdot \beta$.
6. Another server strand s'_s could receive N_b in its original form and transmit N_b and a new session key K'' as $\{\!|B \hat{} K'' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)}$.
7. $\mathsf{used}(S_3) = \{\mathsf{ltk}(A), \mathsf{ltk}(B), K'\}$. Although adding $\mathsf{Lsn}[\mathsf{ltk}(A)]$ and $\mathsf{Lsn}[\mathsf{ltk}(B)]$ lead to dead skeletons, perhaps adding $\mathsf{Lsn}[K']$ does not, i.e. K' may become compromised.

However, we can prune Case 6, because K'' is not usefully different from K'. The adversary cannot use messages transmitted by s'_s differently from the messages transmitted by the existing s_s. In Section 9, we formalize this pruning principle (Prop. 11). Discarding Case 6, there are two live possibilities: either $K' = K$ or else K' becomes compromised. We consider Case 7 next.

Case 7: K' becomes compromised. Consider the skeleton $\mathbb{A}_1 \cup \mathsf{Lsn}[K']$. K' originates uniquely at m_1, so $m_1, (\mathsf{Lsn}[K'] \downarrow 1)$ is an outgoing transformed pair for K' and $S_4 = \{\, \{\!|B \hat{} K' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)}, \{\!|A \hat{} K'|\!\}_{\mathsf{ltk}(B)} \,\}$. Thus, we have the outgoing cohort 8–9:

8. Some role $\mathsf{Init}, \mathsf{Resp}, \mathsf{Serv}$ provides a transforming edge for K', S_4. However, no Yahalom role retransmits it as a subterm of any new message. The initiator uses K' to encrypt a message, but in our model, this discloses nothing. For finer models, see e.g. [3, 5].
9. One of the keys that protects K' in S_4, i.e. a key $K_0 \in \mathsf{used}(S_4)$, becomes compromised; but $\mathsf{used}(S_4) = \{\, \mathsf{ltk}(A), \mathsf{ltk}(B) \,\}$.

So neither Case 8 nor Case 9 is possible. We discard Case 7, as the whole cohort 8–9 is unrealizable or "dead."

Hence, all homomorphisms to realized skeletons must factor through Case 5. Let $\mathbb{A}_2 = \mathbb{A}_1 \cdot \beta$ be the result of replacing K' by K wherever mentioned in $\mathbb{A}_1$. If any homomorphism $H\colon \mathbb{A}_0 \mapsto \mathbb{A}'$ has $\mathbb{A}'$ realized, then H factors through the embedding $\mathbb{A}_0 \mapsto \mathbb{A}_2$.

***B*'s Source for *K*.** The responder B receives $\{\!|A \hat{} K|\!\}_{\mathsf{ltk}(B)}$ on node n_1. We apply the *Incoming Test Principle*, with cohort:

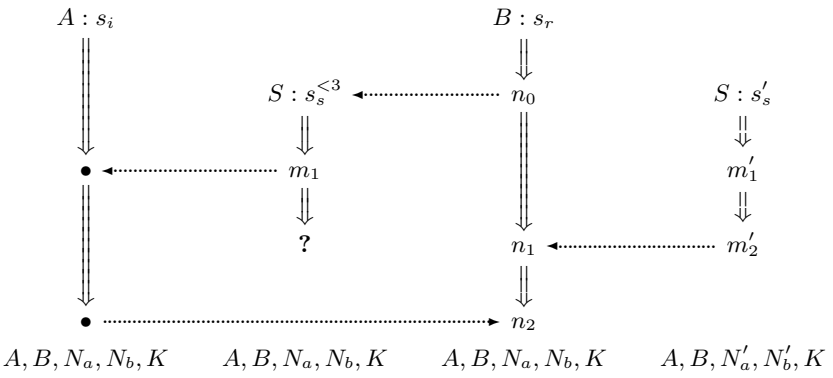

Fig. 4. Preskeleton $\mathbb{A}_3$, with $\mathsf{non}_{\mathbb{A}_3} = \{\mathsf{ltk}(A), \mathsf{ltk}(B)\}$ and $\mathsf{unique}_{\mathbb{A}_3} = \{N_b, K\}$.

10. A server strand s'_s, with parameters A, B, K, transmits $\{\!|A \hat{} K|\!\}_{\mathsf{ltk}(B)}$; possibly different nonces appear in s'_s. The embedding yields $\mathbb{A}_3$ in Fig. 4.
11. Alternatively, $\mathsf{ltk}(B)$ has been compromised and $\{\!|A \hat{} K|\!\}_{\mathsf{ltk}(B)}$ is generated by the adversary. However, $\mathsf{ltk}(B) \in \mathsf{non}_{\mathbb{A}_2}$, excluding this case.

$\mathbb{A}_3$ is not a skeleton, but only a preskeleton, because of an anomaly. $K \in \mathsf{unique}_{\mathbb{A}_3}$ is intended to originate at just one node, but in fact originates at both m_1 and m'_1. Therefore, in any skeleton obtained by a homomorphism $H = [\phi, \alpha]$ jointly

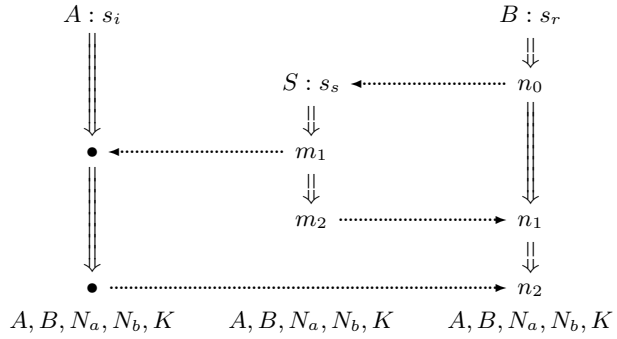

Fig. 5. Skeleton $\mathbb{A}_4$, with $\mathsf{non}_{\mathbb{A}_4} = \{\mathsf{ltk}(A), \mathsf{ltk}(B)\}$ and $\mathsf{unique}_{\mathbb{A}_4} = \{N_b, K\}$.

from the union $\mathbb{A}_2 \cup \{s'_s\} = \mathbb{A}_3$, necessarily $\phi(m_1) = \phi(m'_1)$, equating the strands s_s and s'_s. H must then factor through skeleton $\mathbb{A}_4$ (Fig. 5), where consequently $N_a \cdot \alpha = N'_a \cdot \alpha$ and $N_b \cdot \alpha = N'_b \cdot \alpha$, and the height of $\phi(s_s)$ is 3.

Skeleton $\mathbb{A}_4$ is *realized*: every message received is sent, even without adversary activity. Moreover, $\mathbb{A}_4$ is a *shape*. First, if we leave out any nodes, then either B's

original strand is no longer embedded in the result, or else the result is no longer realized. Second, we cannot make it more general: If two different strands share a parameter, and we alter that parameter in one of the strands, then the result is no longer realized. For instance, the diagram would no longer be realized if A's parameter N_b were altered to some N'_b. Since all homomorphisms from $\mathbb{A}_0$ to realized skeletons factor through $\mathbb{A}_4$, it is the only shape for $\mathbb{A}_0$.

8 Search Strategy

The goal of CPSA is defined using the following notions:

$\mathsf{step}(\mathbb{A}, C)$ which holds if the finite set C of skeletons is an outgoing or incoming cohort for $\mathbb{A}$. Any homomorphism from $\mathbb{A}$ to a realized skeleton passes through some $\mathbb{A}_k \in C$. The principles of Section 6 imply that the tests and their cohorts may be used in any order, while still finding all shapes.

$\mathsf{realized}(\mathbb{A})$ which holds if $\mathbb{A}$ is realized; we can determine this directly.

$\mathsf{min\_real}_{\mathbb{A}_0}(\mathbb{A}')$ which is defined if $\mathbb{A}'$ is realized. Its value is the finite, non-empty set of shapes $\mathbb{A}$ such that (1) there is a homomorphism from $\mathbb{A}_0$ to $\mathbb{A}$; (2) $\mathbb{A}$ is realized; (3) there is a nodewise injective homomorphism from $\mathbb{A}$ to $\mathbb{A}'$; and (4) $\mathbb{A}$ is $\leq_n$-minimal among skeletons satisfying (1–3). $\mathsf{min\_real}_{\mathbb{A}_0}(\mathbb{A}')$ implements the procedure described in the proof of Prop. 8.

We say $\mathsf{child}(\mathbb{A}, \mathbb{A}')$ if for some C, $\mathsf{step}(\mathbb{A}, C)$ and $\mathbb{A}' \in C$. Let $\mathsf{descendent}$ be the reflexive, transitive closure of child. The goal of the search, given a starting skeleton $\mathbb{A}_0$, is to determine the set

$$\mathsf{shapes}(\mathbb{A}_0) = \{\mathbb{A}_2 \colon \exists \mathbb{A}_1 \,.\, \mathsf{descendent}(\mathbb{A}_0, \mathbb{A}_1) \ \wedge \ \mathbb{A}_2 \in \mathsf{min\_real}_{\mathbb{A}_0}(\mathbb{A}_1)\}.$$

To do so, we use the search algorithm in Fig 6. We also need some auxiliaries:

$\mathsf{dead}(\mathbb{A})$ means $\mathbb{A}$ cannot be realized, i.e. there is no realized $\mathbb{A}'$ with $H \colon \mathbb{A} \mapsto \mathbb{A}'$. $\mathsf{Dead}(\mathbb{A})$ follows from any of the following: (1) $\mathbb{A}$ contains $\mathsf{Lsn}[a]$ where $a \in \mathsf{non}_{\mathbb{A}}$; (2) $\mathsf{dead}(\mathbb{A}_0)$ and $H \colon \mathbb{A}_0 \mapsto \mathbb{A}$; or (3) $\mathsf{step}(\mathbb{A}, C)$ where C consists of dead skeletons. Condition (1) was used repeatedly and condition (3) was used to discard Case 7, as the cohort 8–9 consisted of dead skeletons.

$\mathsf{redundant\_strand}(\mathbb{A})$ tests whether $\mathbb{A}$ contains a redundant strand that can be identified with some other strand by a homomorphism from $\mathbb{A}$ to a proper subskeleton. We discarded a redundant strand in Case 6 (see Prop. 11).

$\mathsf{step\_applies}(\mathbb{A})$ tests if an unsolved outgoing or incoming step exists in $\mathbb{A}$.

$\mathsf{apply\_step}(\mathbb{A})$ selects an unsolved step, finds a cohort, updates the step relation, and then returns the cohort (assuming $\mathsf{step\_applies}(\mathbb{A})$ is true).

$\mathsf{targets}(\boldsymbol{H}) = \{\mathbb{A}_k \colon k \leq j\}$, if $\boldsymbol{H}$ is a set of j homomorphisms $H_k \colon \mathbb{A} \mapsto \mathbb{A}_k$.

We assume $\mathsf{select}\ \mathcal{S}$ selects a member of $\mathcal{S}$ if it is non-empty; and $\mathsf{filter}\ p\ \mathcal{S}$ takes the subset of $\mathcal{S}$ satisfying p. The failure marked "Impossible" in Fig. 6 cannot be reached, because completeness (Thm. 3) ensures that when $\mathbb{A}$ is not realized, then some authentication test step applies.

```
F := {A_0};   shapes_found := ∅;   seen := F;
while F ≠ ∅ begin
      A := select(F);   F := F \ {A};
      if realized(A)
        then shapes_found := shapes_found ∪ min_real_{A_0}(A)
      else if redundant_strand(A) then skip
      else if step_applies(A) then begin
            let new = targets(apply_step(A)) \ seen in
            F := F ∪ new;   F := F \ (filter dead F);
            seen := seen ∪ new
            end
      else fail "Impossible."
      end;
return shapes_found
```

Fig. 6. CPSA Search Algorithm

9 Completeness of the Authentication Tests

If a skeleton $\mathbb{A}$ is not realized, does it necessarily contain an outgoing test node or an incoming test node, with no matching transforming node? Yes, it does, and this is the core reason why every shape can be reached with finitely many applications of the authentication tests.

Prop. 4 from Section 4.3 essentially isolates the inductive property of penetrator webs on this depends; we may formalize the conclusion for derivability in skeletons (using Def. 11) as follows.

Proposition 10. *Suppose that for all $a \in \mathsf{unique}_{\mathbb{A}}$, if a originates at any $m \in \mathbb{A}$ and $a \sqsubseteq \mathsf{msg}(n)$, then $m \preceq_{\mathbb{A}} n$. If n_1 is negative and $\mathsf{msg}(n_1)$ is not derivable before n_1 in $\mathbb{A}$, then either:*

1. *(n_0, n_1) is an unsolved outgoing test pair with respect to some a, S:*
   I.e. there exist (i) a node $n_0 \preceq_{\mathbb{A}} n_1$, (ii) an atom $a \in \mathsf{unique}_{\mathbb{A}}$ originating at n_0, (iii) a set S of encryptions such that a occurs only within S in t in positive nodes preceding n_1, and for each $K \in \mathsf{used}(S)$, K^{-1} is not derivable before n_1 in $\mathbb{A}$; or else
2. *n_1 is an unsolved incoming test node for some t, S where $t = \{|t_0|\}_K$:*
   I.e., (i) t occurs only within S in positive nodes before n_1 in $\mathbb{A}$, (ii) t occurs outside S in $\mathsf{msg}(n_1)$, (iii) K is not derivable before n_1 in $\mathbb{A}$, and (iv) for all $K_1 \in \mathsf{used}(S)$ K_1^{-1} is not derivable before n_1 in $\mathbb{A}$.

Proof. If $\mathsf{msg}(n)$ is not derivable before n, we can use Prop. 4, letting $A = \mathsf{non}_{\mathbb{A}} \cup (\mathsf{unique}_{\mathbb{A}} \cap \{a\colon a \text{ originates in } \mathbb{A}\})$. If Prop. 4 Clause 1 (resp. Clause 2) is false, there is an unsolved outgoing test (resp. an unsolved incoming test). □

Shapes, being minimal, do not contain unnecessary listener strands. Thus, a sequence of contractions and augmentations leading to a shape, as in Theorems 1, 2, does not contain listener augmentations. The augmentations it contains are outgoing and incoming augmentations.

Theorem 3 (Authentication Tests Completeness). *Let $J = [\phi_J, \alpha_J] \colon \mathbb{A} \mapsto \mathbb{A}_s$ be a shape. J is isomorphic to $H_i \circ \ldots \circ H_0$ for some sequence of homomorphisms $\{H_j\}_{0 \leq j \leq i}$, where*

1. *For some α' and some subskeleton $\mathbb{A}'$ of $\mathbb{A}$, $H_0 \colon \mathbb{A} \mapsto \mathsf{hull}_{\alpha'}(\mathbb{A}')$ is surjective;*
2. *For each j with $1 \leq j \leq i$, $H_j \colon \mathbb{A}_{j-1} \mapsto \mathbb{A}_j$ is a contraction or an augmentation as in Theorem 1 or Theorem 2, Clauses 1, 2.*

Proof. We define a sequence $\{H_j\}_{0 \leq j \leq i}$ of homomorphisms satisfying (1,2), and a sequence $\{L_j\}_{0 \leq j \leq i}$ such that for every j, (3) L_j is nodewise injective, and (4) $J = L_j \circ H_j \circ \ldots \circ H_0$ (see Fig. 7). By the definition of shape, if any composition $H_j \circ \ldots \circ H_0$ is realized, then L_j is an isomorphism, so we may take $i = j$ and stop.

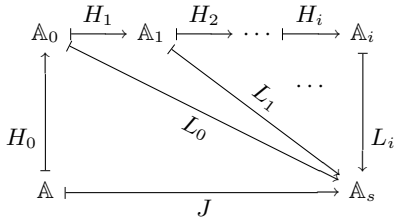

Fig. 7. Augmentations, contractions H_{j+1}, and nodewise injective L_j.

For L_0 to be nodewise injective, with $J = L_0 \circ H_0$, we define H_0 to prune unnecessary strands in $\mathbb{A}$. Partition strands in $\mathbb{A}$ by their image under ϕ_J; i.e.

$$e_s = \{s' \colon \phi_J(s' \downarrow 1) = \phi_J(s \downarrow 1)\}.$$

For each partition element e_s, choose a representative $r(e_s)$ of maximal height. We let β_0 be a most general unifier for all the pairs $\mathsf{msg}(s \downarrow \ell), \mathsf{msg}(s' \downarrow \ell)$ where $e_s = e_{s'}$ and ℓ is no greater than both their heights. Such a mgu β_0 exists, because α_J unifies all these pairs. If ϕ_0 is the map $s \mapsto r(e_s)$, let $H_0 = [\phi_0, \beta_0]$. L_0 is defined to send each $\phi_0(n)$ to $\phi_J(n)$ and each $\beta_0(a)$ to $\alpha_J(a)$.

Next, suppose $H_0 \ldots H_j$ and $L_0 \ldots L_j$ are defined, with $H_j \colon \mathbb{A}_{j-1} \mapsto \mathbb{A}_j$, but $\mathbb{A}_j$ is not realized. Let $L_j = [\psi_j, \beta_j] \colon \mathbb{A}_j \mapsto \mathbb{A}_s$. Choose $n_1 \in \mathbb{A}_j$ be a negative node with $\mathsf{msg}(n_1)$ not penetrator derivable before n_1 in $\mathbb{A}_j$ (Prop. 5).

By Prop. 10, n_1 is an incoming or outgoing test node, for some set of encryptions S and term t, where for an incoming test node $t = \{\!|t_0|\!\}_{K_0}$ and for an outgoing test node $t = a$. However, n_1 has no corresponding transformed node. Consider its image under L_j; there are three essential possibilities.

$\psi_j(n_1)$ **has a corresponding transforming node:** We select a most general preimage of the strand up to the transforming node, and add the preimage strand as an incoming or outgoing augmentation.

$\psi_j(n_1)$ **is no longer a transformed node:** If n_1 is a transformed node relative to S, t but $\psi_j(n_1)$ is not a transformed node relative to $S \cdot \beta_j, t \cdot \beta_j$, then we let H_{j+1} be a most general contraction with this property.

$\psi_j(n_1)$ **has no corresponding transforming node:** By Props. 2, 3, some relevant key K is compromised in the bundle realizing $\mathbb{A}_s$. However, by the choice of n_1, K is not derivable in $\mathbb{A}_j$. Thus, in $\mathbb{A}_s$ there is an outgoing transforming edge for $\mathsf{Lsn}[K \cdot \beta_j]$, and we may augment $\mathbb{A}_j$ with the listener strand $\mathsf{Lsn}[K]$ and a most general preimage of this transforming edge. □

There a couple of fine points to mention about this proof. One is that the most general preimage that we augment with in the first and third case above may be less general than the test node. If so, we first apply a contraction as H_{j+1}, and then do the augmentation as H_{j+2}; see the discussion after Prop. 9. Another fine point concerns the third case. The point of origination for a key K_0 may not yet exist in $\mathbb{A}_j$, but be added in some later augmentation H_k with $k > j$. Possibly $K_0 \notin \mathsf{unique}_{\mathbb{A}_j}$, although $K_0 \cdot \beta_j \in \mathsf{unique}_{\mathbb{A}_s}$. However, K_0 is derivable then, although its image in $\mathbb{A}_s$ is not derivable using only the regular nodes of $\mathbb{A}_s$ that are in the image of $\mathbb{A}_j$. The choice of n_1 ensures that we do not select such a K_0 for the key K in the third case.

A Pruning Condition The nodewise injective sequence $\{L_j\}_{0 \le j \le i}$ in the proof of Thm. 3 ensures that each augmentation in the sequence $\{H_j\}_{0 \le j \le i}$ is actually contributing to the eventual result. No node is added, only to be identified later with any other node on the way to the shape $\mathbb{A}_s$.

By contrast, suppose an augmentation introduces a new strand that *could* be identified with an existing strand. The following proposition shows that it *will* be identified with some existing strand before a shape is reached. Thus, we do not miss any shapes if we never add a strand that is redundant in this sense.

Proposition 11. *Let $\mathbb{A}$ be a subskeleton of $\mathbb{A}'$ with idempotent $I = [\psi, \beta] \colon \mathbb{A}' \mapsto \mathbb{A}$. If $J' = [\phi', \alpha'] \colon \mathbb{A}' \mapsto \mathbb{A}'_s$ is a shape, then $\psi(m) = n$ implies $\phi'(m) = \phi'(n)$.*

Proof. Since $\mathbb{A}$ is a subskeleton of $\mathbb{A}'$, there is an embedding $E \colon \mathbb{A} \mapsto \mathbb{A}'$. Let

$$\begin{array}{ccccccc}
\mathbb{A}' & \xrightarrow{H'_0} & \mathbb{A}'_0 & \xrightarrow{H'_1} & \cdots & \xrightarrow{H'_i} & \mathbb{A}'_i = \mathbb{A}'_s \\
E \uparrow & & E_0 \uparrow & & & & E_i \uparrow \\
\mathbb{A} & \xrightarrow{H_0} & \mathbb{A}_0 & \xrightarrow{H_1} & \cdots & \xrightarrow{H_i} & \mathbb{A}_i
\end{array}$$

Fig. 8. Homomomorphisms and embeddings justifying pruning

$H'_i \circ \ldots \circ H'_0$, as shown in Fig. 8, be the decomposition of J' as in Thm. 3, with successive targets $\mathbb{A}'_j$. We construct a corresponding sequence $H_i \circ \ldots \circ H_0 \colon \mathbb{A} \mapsto \mathbb{A}_i$, together with a sequence of embeddings $E_j \colon \mathbb{A}_j \mapsto \mathbb{A}'_j$. For H_0, we identify any two strands in $\mathbb{A}_0$ if they are identified by H'_0, and contract atoms mentioned in $\mathbb{A}_0$ if they are contracted by H'_0. If H'_{j+1} augments or contracts to solve a node $n_1 \in \mathbb{A}_j$, then H_{j+1} applies the same augmentation or contraction. If $n_1 \in \mathbb{A}'_j \setminus \mathbb{A}_j$, then we let H_{j+1} be the identity. Evidently, we can construct E_{j+1} from E_j since augmentations to $\mathbb{A}_j$ have also been applied to $\mathbb{A}'_j$.

Derivability is preserved, since for any node in $n \in \mathbb{A}_j$ whose derivation in $\mathbb{A}'_j$ uses a positive node $m \in \mathsf{nodes}_{\mathbb{A}'_j} \setminus \mathsf{nodes}_{\mathbb{A}_j}$, we use $\psi(m)$ instead.

Thus, the target $\mathbb{A}_i$ of $H_i \circ \ldots \circ H_0$ is a realized substructure of $\mathbb{A}'_s$. So $H_i \circ \ldots \circ H_0 \circ I \colon \mathbb{A}'_0 \mapsto \mathbb{A}_i$. Since the embedding $E_i \colon \mathbb{A}_i \mapsto \mathbb{A}'_s$ is node injective, and J' is a shape, E_i is an isomorphism. □

10 Implementing CPSA

We discuss here two aspects of the CPSA implementation. They are: (1) finding candidate transforming edges in protocols, and using unification in applying them; and (2) choosing sets S for outgoing tests, and representing the sets.

Finding transforming edges. When CPSA reads a protocol description in its input format, it identifies all the potential transforming edges. For the outgoing tests, it locates all candidate pairs of a reception node m_0 and a transmission node m_1 later on the same role such that a key or nonce is received in one or more encrypted forms on m_0 and retransmitted outside these forms in m_1. For incoming tests, CPSA notes all transmission nodes m_1 that send encrypted units.

To find outgoing transforming edges for $a \in \mathsf{unique}_{\mathbb{A}}$ and a set S, CPSA considers each candidate edge $m_0 \Rightarrow^+ m_1$. Suppose an encrypted sub-message t of $\mathsf{msg}(m_0)$ unifies with a member of S using a replacement α. If $a \cdot \alpha$ occurs in $\mathsf{msg}(m_0) \cdot \alpha$, but only within $S \cdot \alpha$, then we check $\mathsf{msg}(m_1) \cdot \alpha$. If it occurs outside $S \cdot \alpha$ in $\mathsf{msg}(m_1) \cdot \alpha$, then $m_0 \Rightarrow^+ m_1$ is a successful candidate. If α contracts atoms, then we apply the Outgoing Test Principle twice, once to apply this contraction, and once to add the instance of $m_0 \Rightarrow^+ m_1$.[3] We also check whether a contraction eliminates the outgoing test edge entirely, as in Case 5.

For incoming tests, we do a unification on the candidate nodes m_1.

Selecting sets S for outgoing tests. To select sets S in the outgoing test principle, we use a trick we call the "forwards-then-backwards" technique. CPSA plans a sequence of applications of the outgoing test until no further transforming edge is found, as in Yahalom cases 3 and 1. It follows the transmission of the uniquely originating value—N_b in that case—forwards. Newly introduced atoms

[3] This and the incoming test with $S \neq \emptyset$ are the only aspects of the authentication test search that do not occur in the Yahalom analysis.

like K' are implicitly universal. Originally, N_b occurs only in $\{\!|A \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(B)}$; after a server strand it also occurs in $\{\!|B \hat{} K' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)}$. After an initiator strand, no other transforming edges can succeed.

CPSA uses the sets in the opposite order. The set $S_1 = \{\{\!|A \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(B)}\} \cup \{\{\!|B \hat{} K' \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(A)} : K' \text{ a key}\}$ is used first to introduce the initiator transforming edge. Then the smaller set $\{\{\!|A \hat{} N_a \hat{} N_b|\!\}_{\mathsf{ltk}(B)}\}$ is used to introduce the (earlier) server transforming edge.[4]

The forwards-then-backwards technique suggested CPSA's representation for the sets S. These sets are not necessarily finite; S_1 e.g. is not. The family is closed under union and set difference. The primitive members are singletons $\{t_0\}$ and sets that represent all the instances of a term t_1 as some of t_1's parameters vary. Thus, we can represent all candidate sets are as finite unions and differences of values of the form $\lambda \boldsymbol{v} \,.\, t$, where the vector $\boldsymbol{v}$ binds 0 or more atoms in t. Completeness requires only sets S representable in this form.

This representation fits also nicely with our use of unification to provide an extremely focused search, leading to good runtimes on a variety of protocols. Samples run on a Thinkpad X31, with a 1.4 GHz Pentium M processor and 1 GB store, under Linux, are shown in Fig. 9. CPSA is implemented in OCaml.

Protocol	Point of view	Runtime	Shapes
ISO reject	responder	0.193s	2
Kerberos	client	1.443s	1
Needham-Schroeder	responder	0.055s	1
Needham-Schroeder-Lowe	responder	0.124s	1
Yahalom	responder	2.709s	1

Fig. 9. Protocols with CPSA runtimes

Conclusion

In this paper, we have developed the theory of skeletons and homomorphisms. We used it together with the strong form of the authentication tests (Props. 2–3) to establish search-oriented versions of the tests (Thms. 1–2). We described how to use these ideas to mechanize protocol analysis. Finally, we showed that these tests have a form of *completeness* (Thm. 3).

The soundness of the search algorithm does not require the bare-bones Dolev-Yao model used here. One can augment CPSA with Diffie-Hellman operations, as studied for an earlier strand-based method in [11]. One can also allow keys to be complex messages, typically the result of hashing. In our current framework, replacements map atoms to other atoms only, but it should be possible to map

[4] The cleverer set S_2 we used in Case 3 is an optimization. To ensure that the server and initiator agree on the session key, CPSA uses instead a cohort similar to Cases 6–7.

atoms to terms in general, at the cost of using more sophisticated methods to check whether skeletons are realized (e.g. [15]). Indeed, the skeletons-and-homomorphisms approach may remain useful in a cryptographic, asymptotic probabilistic context.

Acknowledgments. We thank Lenore Zuck and John D. Ramsdell for their comments. Larry Paulson suggested the Yahalom protocol as a challenge.

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. Michael Backes and Birgit Pfitzmann. Relating cryptographic and symbolic key secrecy. In *Proceedings, 26th IEEE Symposium on Security and Privacy*, May 2005. Extended version, `http://eprint.iacr.org/2004/300`.
4. Bruno Blanchet and Andreas Podelski. Verification of cryptographic protocols: Tagging enforces termination. In *Foundations of Software Science and Computation Structures*, LNCS, pages 136–152, April 2003.
5. Ran Canetti and Jonathan Herzog. Universally composable symbolic analysis of mutual authentication and key exchange protocols. In *Proceedings, Theory of Cryptography Conference (TCC)*, March 2006.
6. 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`.
7. Daniel Dolev and Andrew Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.
8. 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.
9. Andrew D. Gordon and Alan Jeffrey. Types and effects for asymmetric cryptographic protocols. *Journal of Computer Security*, 12(3/4):435–484, 2003.
10. 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.
11. Jonathan C. Herzog. The Diffie-Hellman key-agreement scheme in the strand-space model. In *16th Computer Security Foundations Workshop*, pages 234–247, Asilomar, CA, June 2003. IEEE CS Press.
12. ITU. Message sequence chart (MSC). Recommendation Z.120, 1999.
13. Leslie Lamport. Time, clocks and the ordering of events in a distributed system. *CACM*, 21(7):558–565, 1978.
14. 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.
15. 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.

16. Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12), 1978.
17. Lawrence C. Paulson. Relations between secrets: Two formal analyses of the Yahalom protocol. *Journal of Computer Security*, 2001.
18. Adrian Perrig and Dawn Xiaodong Song. Looking for diamonds in the desert: Extending automatic protocol generation to three-party authentication and key agreement protocols. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, July 2000.
19. R. Ramanujam and S. P. Suresh. Decidability of context-explicit security protocols. *Journal of Computer Security*, 13(1):135–166, 2005. Preliminary version appeared in WITS '03, *Workshop on Issues in the Theory of Security*, Warsaw, April 2003.