CPSA Theory

Moses D. Liskov John D. Ramsdell
Paul D. Rowe
The MITRE Corporation
CPSA Version 2.2.5

August 16, 2011

This is a draft and there is much that is missing. We expect this document is in for a big change.

CPSA takes a partial description of a run of a protocol, and attempts to produce a compact description of all possible runs of the protocol compatible with the partial description. Given a partial description, CPSA uses an authentication test to infer what else must have happened, and thereby reduce the problem to finding possible runs starting with a set of more refined descriptions. The goal of this document is to precisely describe authentication tests.

The formal definition of a partial run of a protocol is called a skeleton, and is introduced in Section 4. To motivate the definition, Section 1 describes a simplified version of a message algebra used in CPSA. Section 2 describes a bundle [4, 2], a model of asynchronous messages-passing that includes the behaviors of honest and adversarial participants. It also introduces the notion of a protocol, and specifies what it means for a bundle to be a run of a protocol.

Section 3 describes the capabilities of the adversary. CPSA does not explicitly represents adversarial behaviors. Section 4 and Section 5 reveal the means by which the details of adversarial behavior are abstracted away. Finally, Section 6 describes authentication tests.

Sorts: NAME, TEXT, DATA, SKEY, AKEY < MESG

Base sorts: NAME, TEXT, DATA, SKEY, AKEY

Carried positions: $\bullet$ denotes a carried position.

$\{\!\vert \bullet \vert\!\}_{(\cdot)}\colon$ MESG $\times$ MESG $\to$ MESG	Encryption
$(\bullet, \bullet)\colon$ MESG $\times$ MESG $\to$ MESG	Pairing
"…": MESG	Tag constants
$K_{(\cdot)}\colon$ NAME $\to$ AKEY	Public key of name
$(\cdot)^{-1}\colon$ AKEY $\to$ AKEY	Inverse of key
$\mathsf{ltk}(\cdot, \cdot)\colon$ NAME $\times$ NAME $\to$ SKEY	Long term key

Equation: $(x^{-1})^{-1} \approx x$ for $x\colon$ AKEY

Figure 1: Basic Crypto Signature and Equation

1 Order-Sorted Message Algebras

CPSA models a message by an equivalence class of terms over a signature. A sort system is used to classify messages. CPSA depends on the sort system to allow it to treat a variable that represents an asymmetric key differently from a variable that represents an arbitrary message. In particular, CPSA uses order-sorted quotient term algebras [1] for message algebras. This formalism enables the use of well-known algorithms for unification and matching in the presences of equations and sorts [3, Chapter 8].

This paper makes no attempt to provide a general introduction to order-sorted quotient term algebras. We use a message algebra called the Basic Crypto Algebra (BCA), which is the main algebra used by CPSA.

There are six BCA sorts: MESG, the sort of all messages, SKEY, the sort of symmetric keys, AKEY, the sort of asymmetric keys, NAME, the sort of participant names, and TEXT and DATA for ordinary values. All sorts are subsorts of MESG. The function symbols, or *operations*, used to form terms are given by the signature in Figure 1.

Each variable x used to form a term has a unique sort s, written $x\colon s$. Variable set X is an indexed set of sets of variables, $X_s = \{x \mid x\colon s\}$. For BCA, X_{MESG}, X_{SKEY}, X_{AKEY}, X_{NAME}, X_{TEXT}, and X_{DATA} partition the set of variables in X. By abuse of notation, at times, we write X for the set of variables in X.

The Basic Crypto Quotient Term Algebra $\mathfrak{A}$ generated by variable set X

$$
\begin{aligned}
\mathfrak{A}_{\mathrm{SKEY}} &= \{\{x\} \mid x \in X_{\mathrm{SKEY}}\} \cup \{\{\mathsf{ltk}(a,b)\} \mid a \in X_{\mathrm{NAME}}, b \in X_{\mathrm{NAME}}\}\\
\mathfrak{A}_{\mathrm{AKEY}} &= \{\{x^{-2n} \mid n \in \mathbb{N}\} \mid x \in X_{\mathrm{AKEY}}\}\\
&\quad \cup \{\{x^{-2n-1} \mid n \in \mathbb{N}\} \mid x \in X_{\mathrm{AKEY}}\}\\
&\quad \cup \{\{K_x^{-2n} \mid n \in \mathbb{N}\} \mid x \in X_{\mathrm{NAME}}\}\\
&\quad \cup \{\{K_x^{-2n-1} \mid n \in \mathbb{N}\} \mid x \in X_{\mathrm{NAME}}\}\\
\mathfrak{A}_{\mathrm{NAME}} &= \{\{x\} \mid x \in X_{\mathrm{NAME}}\}\\
\mathfrak{A}_{\mathrm{TEXT}} &= \{\{x\} \mid x \in X_{\mathrm{TEXT}}\}\\
\mathfrak{A}_{\mathrm{DATA}} &= \{\{x\} \mid x \in X_{\mathrm{DATA}}\}\\
\textsc{Tags} &= \{\{x\} \mid x \text{ is a tag constant }\}\\
\mathfrak{B} &= \mathfrak{A}_{\mathrm{SKEY}} \cup \mathfrak{A}_{\mathrm{AKEY}} \cup \mathfrak{A}_{\mathrm{NAME}} \cup \mathfrak{A}_{\mathrm{TEXT}} \cup \mathfrak{A}_{\mathrm{DATA}}\\
\mathfrak{A}^0 &= \mathfrak{B} \cup \{\{x\} \mid x \in X_{\mathrm{MESG}}\} \cup \textsc{Tags}\\
\mathfrak{A}^{n+1} &= \mathfrak{A}^n \cup \{\{(t_0, t_1) \mid t_0 \in T_0, t_1 \in T_1\} \mid T_0 \in \mathfrak{A}^n, T_1 \in \mathfrak{A}^n\}\\
&\quad \cup \{\{\{\!|t_0|\!\}_{t_1} \mid t_0 \in T_0, t_1 \in T_1\} \mid T_0 \in \mathfrak{A}^n, T_1 \in \mathfrak{A}^n\}\\
\mathfrak{A} = \mathfrak{A}_{\mathrm{MESG}} &= \textstyle\bigcup_{n\in\mathbb{N}} \mathfrak{A}^n
\end{aligned}
$$

Figure 2: BCA Messages $\mathfrak{A}$ and Atoms $\mathfrak{B}$

is displayed in Figure 2. The union of the messages in $\mathfrak{A}$ is set of terms generated by X, and $\mathfrak{A}$ partitions the set of terms into a set of equivalence classes induced by the equations. Terms t_0 and t_1 are equivalent, written $t_0 \equiv t_1$, iff $t_0 \in T \wedge t_1 \in T$ for some $T \in \mathfrak{A}$. The canonical representative of a message is the t in $\{t' \mid t' \equiv t\}$ with the fewest occurrences of the $(\cdot)^{-1}$ operation.

Keys, names, data, and texts in the algebra are called *atoms* and are members of $\mathfrak{B}$. We write $t\colon \mathfrak{B}$ iff $t\colon S$ for some $S \neq$ MESG. Note that encryption is defined with an encryption key of sort MESG. When the encryption key is of sort AKEY this is meant to model asymmetric encryption: otherwise, this models symmetric encryption. Note that even complex messages such as encryptions can be used as encryption keys in the symmetric sense.

To find the decryption key associated with an encryption, one must exclude the case in which the key is a variable of sort MESG, as there is no way to determine if the encryption operation denotes symmetric or asymmetric encryption. Therefore, the decryption key associated with encryption key t is $inv(t)$.

$$inv(t) = \begin{cases} \mathsf{invk}(t) & \text{if } t\colon \text{AKEY};\\ \text{undefined} & \text{if } t \text{ is a variable of sort } \top;\\ t & \text{otherwise.} \end{cases}$$

An important property possessed by the algebra is that for all $T \in \mathfrak{A}$, if there are any encryptions in T then all members of T are encryptions. As a result, a message can be identified as representing an encryption and if it is, decomposed into its plaintext and its decryption key. This property is a consequence of the fact that equations relate atoms, not arbitrary messages. A similar property holds for pairs.

We write $\mathfrak{A}_X$ when it is important to identify the variable set X that generates the algebra. Given two variable sets X and Y, a *substitution* is an order-sorted map $\sigma\colon X \to \mathfrak{A}_Y$ such that $\sigma(x) \neq x$ for only finitely many elements of X. For a substitution σ, the *domain* is the set of variables $Dom(\sigma) = \{x \mid \sigma(x) \neq x\}$ and the *range* is the set $Ran(\sigma) = \{\sigma(x) \mid x \in Dom(\sigma)\}$. Substitution σ_0 is *more general than* σ_1, written $\sigma_0 \trianglelefteq \sigma_1$, if there exists a substitution σ_2 such that $\forall x\, \sigma_1(x) \equiv \sigma_2(\sigma_0(x))$. Given a substitution $\sigma\colon X \to \mathfrak{A}_Y$, the unique homomorphism $\sigma^*\colon \mathfrak{A}_X \to \mathfrak{A}_Y$ induced by σ is also denoted σ.

In what follows, a finite sequence is a function from an initial segment of the whole numbers. The length of a sequence f is $|f|$, and sequence $f = \langle f(1), \ldots, f(n)\rangle$ for $n = |f|$. Alternatively, $\langle x_1, x_2, \ldots, x_n\rangle = x_1 :: x_2 :: \ldots :: x_n :: \langle\rangle$. If S is a set, then S^* is the set of finite sequences of S, and S^+ is the non-empty finite sequences of S.

The concatenation of sequences f_0 and f_1 is $f_0 \frown f_1$. When the context distinguishes sequences and their elements, such as for sequences of integers, we often write $f_0 \frown 1 \frown f_1$ instead of $f_0 \frown \langle 1\rangle \frown f_1$. The prefix of sequence f of length n is $f|_n$.

A *position* p is a finite sequence of whole numbers. The term in t that *occurs at* p, written $t \mathbin{@} p$, is:

$$\begin{aligned} &t \mathbin{@} \langle\rangle = t;\\ &(t_1, t_2) \mathbin{@} i :: p = t_i \mathbin{@} p \text{ for } i \in \{1, 2\};\\ &\{\!|t_1|\!\}_{t_2} \mathbin{@} i :: p = t_i \mathbin{@} p \text{ for } i \in \{1, 2\};\\ &t^{-1} \mathbin{@} 1 :: p = t \mathbin{@} p. \end{aligned}$$

A term t *occurs in* term t' if $t = t' \mathbin{@} p$ for some p. A message T *occurs in* message T' if the canonical representative of T occurs in the canonical representative of T'.

A carried term is one that can be extracted from a message reception assuming plaintext is extractable from encryptions. The positions at which term t is carried in t' is $\mathit{carpos}(t, t')$, where

$$\mathit{carpos}(t,t') = \begin{cases} \{\langle\rangle\} & \text{if } t' \equiv t, \text{ else} \\ \{1 :: p \mid p \in \mathit{carpos}(t,t_1)\} & \\ & \text{if } t' = \{\!|t_1|\!\}_{t_2}, \text{ else} \\ \{i :: p \mid i \in \{1,2\}, p \in \mathit{carpos}(t,t_i)\} & \\ & \text{if } t' = (t_1,t_2) \text{ else} \\ \emptyset & \text{otherwise.} \end{cases}$$

Term t *carries* t' if $\mathit{carpos}(t', t)$ is not empty, and $t' \sqsubseteq t$ when t' is carried by t. Note that for all terms t_0, t_1, t'_0, t'_1, if $t_0 \equiv t_1$ and $t'_0 \equiv t'_1$, then $\mathit{carpos}(t_0, t'_0) = \mathit{carpos}(t_1, t'_1)$. We write $t' \sqsubseteq_p t$ when $p \in \mathit{carpos}(t', t)$ and $t \mathbin{@} p \equiv t'$.

In what follows, we will often conflate a term with the message of which it is a member, and use lowercase letters to denote both.

2 Strand Spaces and Bundles

A run of a protocol is viewed as an exchange of messages by a finite set of local sessions of the protocol. Each local session is called a *strand*. The behavior of a strand, its *trace*, is a sequence of messaging events. An *event* is either a message transmission or a reception. Outbound message $t \in \mathfrak{A}_X$ is written as $+t$, and inbound message t is written as $-t$. The set of traces over $\mathfrak{A}_X$ is $\mathfrak{C}_X = (\pm\mathfrak{A}_X)^+$. A message *originates* in a trace if it is carried by some event and the first event in which it is carried is outbound. A message is *gained* by a trace if it is carried by some event and the first event in which it is carried is inbound. A message is *acquired* by a trace if it first occurs in a reception event and is also carried by that event.

Abstractly, a strand space is a multiset of traces, but since we wish to name each element, a *strand space* Θ_X over algebra $\mathfrak{A}_X$ is defined to be a sequence of traces in $\mathfrak{C}_X$. A strand s is a member of the domain of Θ_X, and its trace is $\Theta_X(s)$. In a strand space, the elements of the generator set X denote atomic message elements, such as keys, and not composite messages, such as encryptions and pairs. Therefore, the sort of every variable in X is a base sort.

Message events occur at nodes in a strand space. For each strand s, there is a node for every event in $\Theta(s)$. The *nodes* of strand space Θ are $\{(s,i) \mid s \in \mathit{Dom}(\Theta), 1 \leq i \leq |\Theta(s)|\}$, the event at a node is $\mathit{evt}_\Theta(s,i) = \Theta(s)(i)$, and the message at a node is $\mathit{msg}_\Theta(s,i) = m$ such that $\mathit{evt}_\Theta(s,i) = \pm m$. Just as a position names a subterm within another term, a strand names a trace within a strand space, and a node names an event in a strand space. The relation $\Rightarrow$ defined by $\{(s,i) \Rightarrow (s,i+1) \mid s \in \mathit{Dom}(\Theta), 1 \leq i < |\Theta(s)|\}$ is called the *strand succession relation.*

A *bundle* in strand space Θ is a finite directed acyclic graph $\Upsilon(\Theta, \rightarrow)$, where the vertices are the nodes of Θ, and an edge represents communication ($\rightarrow$) or strand succession ($\Rightarrow$). For communication, if $n_0 \rightarrow n_1$, then there is a message t such that $\mathit{evt}_\Theta(n_0) = +t$ and $\mathit{evt}_\Theta(n_1) = -t$. For each reception node n_1, there is a unique transmission node n_0 with $n_0 \rightarrow n_1$.

Each acyclic graph has a transitive asymmetric relation $\prec$ on its vertices. The relation specifies the causal ordering of nodes in a bundle. Relation R on set S is *asymmetric* iff $x\,R\,y$ implies not $y\,R\,x$ for all distinct $x, y \in S$.

An atom *uniquely originates* in a bundle if it originates in the trace of exactly one strand. An atom is *non-originating* in a bundle if it originates on no strand, but each of its variables occurs in some strand's trace.

In a run of a protocol, the behavior of each strand is constrained by a role in a protocol. Adversarial strands are constrained by roles as are non-adversarial strands. A protorole over $\mathfrak{A}_Y$ is $\mathsf{r}_Y(C,N,U)$, where $C \in \mathfrak{C}_Y$, $N \subseteq \mathfrak{B}_Y$, and $U \subseteq \mathfrak{B}_Y$. The trace of the role is C, its non-origination assumptions are N, and its unique origination assumptions are U. A protorole is a *role* if (1) $t \in N$ implies t is not carried in C, and all variables in N occur in C, (2) $t \in U$ implies t originates in C, and (3) if variable x occurs in C then x is an atom or it is acquired in C. A *protocol* is a set of roles. Let $\mathit{Vars}(P)$ be the set of variables that occur in the traces of the roles in protocol P.

A bundle $\Upsilon(\Theta_X, \rightarrow)$ is a *run of protocol* P if there is a role mapping $\mathit{rl}\colon \Theta_X \rightarrow P$ that satisfies properties for each $s \in \mathit{Dom}(\Theta_X)$. Assuming $\mathit{rl}(s) = \mathsf{r}_Y(C,N,U)$ and X and Y share no variables, and let $h = |\Theta_X(s)|$, the properties are (1) $h \leq |C|$, (2) there is a homomorphism $\sigma\colon \mathfrak{A}_Y \rightarrow \mathfrak{A}_X$ such that $\sigma \circ C|_h = \Theta_X(s)$, (3) $\mathit{Dom}(\sigma)$ is the set of variables that occur in $C|_h$, (4) if the variables in $t \in N$ occur in $\mathit{Dom}(\sigma)$, then $\sigma(t)$ is non-originating in $\Upsilon(\Theta_X, \rightarrow)$, and (5) if $t \in U$ originates at index i in C, and $i \leq h$, then $\sigma(t)$ uniquely originates in $\Upsilon(\Theta_X, \rightarrow)$ at node (s,i). Origination assumptions in bundles specified by roles are called *inherited origination assumptions.*

Create($z\colon \mathfrak{B}$)	$\langle +z\rangle$	$\langle +\text{“}\ldots\text{”}\rangle$
Pair($x, y\colon$ MESG)	$\langle -x, -y, +(x,y)\rangle$	$\langle -(x,y), +x, +y\rangle$
Encrypt($x, y\colon$ MESG)	$\langle -x, -y, +\{\!\|x\|\!\}_y\rangle$	$\langle -\{\!\|x\|\!\}_y, -\mathit{inv}(y), +x\rangle$

Figure 3: Basic Crypto Algebra Penetrator Role Traces

3 Adversary Model

A fixed set of penetrator roles encodes the adversary model associated with a message algebra. For the Basic Crypto Algebra, there are eight roles. Each role makes no origination assumptions, and the trace of each role is given in Figure 3. The first line of the figure specifies five traces, one for base sort, and a trace for each tag.

A strand exhibits non-adversarial behavior when its role is not a penetrator role. A non-adversarial strand is called a *regular* strand as is its role.

The penetrator cannot use a non-originating atom to encrypt or decrypt a message, because every key it uses must be carried in a message. Consider a uniquely originating atom that originates on a regular strand. The penetrator cannot make the atom using a create role, because the atom would originate at more than one node. Therefore, the penetrator can use a uniquely originating atom to encrypt or decrypt a message only if it is transmitted by a regular strand unprotected by encryption.

4 Skeletons

The details of penetrator behavior are abstracted away when performing protocol analysis. The abstracted description of a bundle is called a realized skeleton, which is defined using a protoskeleton. A *protoskeleton* over $\mathfrak{A}_X$ is $\mathsf{k}_X(\mathit{rl}, P, \Theta_X, \prec, N, U)$, where $\mathit{rl}\colon \Theta_X \to P$ is a role map, the sets X and $\mathit{Vars}(P)$ are disjoint, Θ_X is a sequence of traces in $\mathfrak{C}_X$, $\prec$ is a relation on the nodes in Θ_X, $N \subseteq \mathfrak{B}_X$ are its non-origination assumptions, and $U \subseteq \mathfrak{B}_X$ are its unique origination assumptions. Unlike a strand space, the sort of a variable in X need not be a base sort.

Assume the strands in bundle $\Upsilon(\Theta_X, \to)$ have been permuted so that regular strands precede penetrator strands in sequence Θ_X, and rl demonstrates the bundle is a run of protocol P. Let P' be P without penetrator roles. Skeleton $\mathsf{k}_X(rl', P', \Theta'_X, \prec, N, U)$ *realizes* the bundle if rl' and Θ'_X are the truncations of rl and Θ_X respectively that omit penetrator strands from their domains, $\prec$ is the transitive asymmetric relation associated with the bundle without penetrator nodes, N is the set of non-originating atoms with variables that occur in Θ'_X, and U is the set of atoms that uniquely originate and are carried by some regular node.

A protoskeleton $\mathsf{k}_X(rl, P, \Theta_X, \prec, N, U)$ is a *preskeleton* if the following properties hold.

1. Sequence rl demonstrates that the strands in $Dom(\Theta_X)$ satisfy the conditions for being a part of a run of protocol P.

2. Relation $\prec$ is transitive, asymmetric, and includes the strand succession relation ($\Rightarrow$).

3. Each atom in N is carried at no node, and each variable in the atom occurs at some node.

4. Each atom in U is carried at some node.

5. N includes the non-originating atoms inherited from roles via the role map.

6. U includes the uniquely originating atoms inherited from roles via the role map.

Let $\mathcal{O}_k(t)$ be the set of nodes at which t originates in preskeleton k, and $\mathcal{G}_k(t)$ be the set of nodes at which t is gained in k. Preskeleton $\mathsf{k}_X(rl, P, \Theta_X, \prec, N, U)$ is a *skeleton* if each atom in U originates on at most one strand, and the node of origination precedes each node that gains the atom, i.e. for every $t \in U$, $n_0 \in \mathcal{O}_k(t)$ and $n_1 \in \mathcal{G}_k(t)$ implies $n_0 \prec n_1$.

Let $k_0 = \mathsf{k}_X(rl_0, P, \Theta_0, \prec_0, N_0, U_0)$ and $k_1 = \mathsf{k}_Y(rl_1, P, \Theta_1, \prec_1, N_1, U_1)$ be preskeletons. There is a *proto-homomorphism* from k_0 to k_1 if ϕ and σ are maps with the following properties:

1. ϕ maps strands of k_0 into those of k_1, and nodes as $\phi((s, p)) = (\phi(s), p)$, that is ϕ is in $Dom(\Theta_0) \rightarrow Dom(\Theta_1)$;

2. $\sigma \colon \mathfrak{A}_X \rightarrow \mathfrak{A}_Y$ is a message algebra homomorphism;

3. $n \in nodes(\Theta_0)$ implies $\sigma(evt_{\Theta_0}(n)) = evt_{\Theta_1}(\phi(n))$;

4. $\sigma(N_0) \subseteq N_1$;

5. $\sigma(U_0) \subseteq U_1$;

A proto-homomorphism is *structure-preserving* if $n_0 \prec_0 n_1$ implies $\phi(n_0) \prec_1 \phi(n_1)$. We write $k_0 \overset{\phi,\sigma}{\longmapsto} k_1$ when (ϕ, σ) is structure-preserving. A proto-homomorphism is a *preskeleton homomorphism* if it is structure-preserving and also, $t \in U_0$ implies $\phi(\mathcal{O}_{k_0}(t)) \subseteq \mathcal{O}_{k_1}(\sigma(t))$, that is, the node at which each uniquely originating atom originates is preserved under homomorphism.

A homomorphism is *strandwise injective* if its strand map is injective. Two preskeletons are isomorphic if they are related by strandwise injective homomorphism in both directions. A homomorphism is *nodewise isomorphic* if the strand map ϕ implies a bijection on nodes, and $n_0 \prec_1 n_1$ implies $\phi^{-1}(n_0) \prec_0 \phi^{-1}(n_1)$. A skeleton is *realized* if there is a nodewise isomorphic homomorphism from it to a skeleton that realizes a bundle, and message component of the homomorphism is injective.

Our formalism requires that every protocol include a listener role of the form: $\mathit{lsn}(x\colon \top) = \mathsf{r}(\langle -x, +x\rangle, \emptyset, \emptyset)$. Instances of this role are sometimes used to make penetrator derived messages visible in skeletons. We say skeleton k *realizes modulo listeners* bundle $\Upsilon(\Theta, \to)$ if k realizes $\Upsilon(\Theta', \to')$ and $\Upsilon(\Theta, \to)$ is the result of removing full length listener strands, and adjusting the communication ordering $\to$ appropriately.

The set of bundles denoted by preskeleton k, $[\![k]\!]$, is:

$$[\![k_0]\!] = \{\Upsilon \mid k_0 \overset{\phi,\sigma}{\longmapsto} k_1 \text{ and } k_1 \text{ realizes modulo listeners } \Upsilon\}$$

A CPSA algorithm is *complete* if when given a preskeleton k_0, either the algorithm diverges, or else it terminates and produces a finite set of realized skeletons K, such that $[\![k_0]\!] = \bigcup_{k_1 \in K} [\![k_1]\!]$.

Let $\longrightarrow$ be an irreflexive reduction relation on preskeletons. The relation $\longrightarrow$ is *semantics preserving* if $[\![k_0]\!] = \bigcup_{k_1 \in \{k_1 \mid k_0 \longrightarrow k_1\}} [\![k_1]\!]$.

4.1 Dolev-Yao Example 1.3

The example has an initiator and responder role.

$$\begin{aligned} \mathit{init}(a, b\colon A, m\colon S) &= \mathsf{r}(\langle +\{\!|\{\!|m|\!\}_b, a|\!\}_b, -\{\!|\{\!|m|\!\}_a, b|\!\}_a\rangle, \emptyset, \emptyset) \\ \mathit{resp}(a, b\colon A, m\colon \top) &= \mathsf{r}(\langle -\{\!|\{\!|m|\!\}_b, a|\!\}_b, +\{\!|\{\!|m|\!\}_a, b|\!\}_a\rangle, \emptyset, \emptyset) \end{aligned}$$

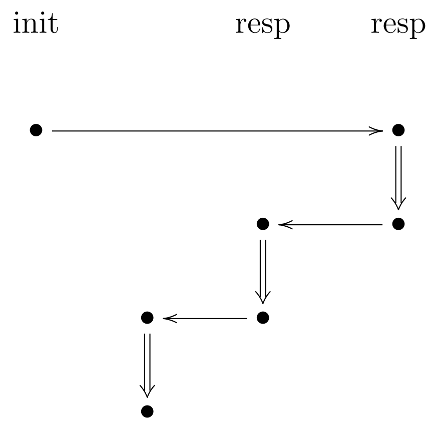

Figure 4: Dolev-Yao Example 1.3 Shape

The algebra for the initiator is generated from X, where $X_\top = \emptyset$, $X_S = \{m\}$, and $X_A = \{a, b\}$, and the algebra for the responder is generated from Y, where $Y_\top = \{m\}$, $Y_S = \emptyset$, and $Y_A = \{a, b\}$,

An interesting point of view for analysis is to see if m is kept secret after the initiator sends its message. Let variable set $Z = a, b\colon A, m\colon S$. The initial scenario preskeleton is:

$$
\begin{array}{ll}
\mathsf{k}_Z(\langle \mathit{init}(a_0, b_0, m_0), \mathit{lsn}(x)\rangle, & \text{Role map} \\
\quad \{\mathit{init}(a_0, b_0, m_0), \mathit{resp}(a_1, b_1, m_1), \mathit{lsn}(x)\}, & \text{Protocol} \\
\quad \langle\langle +\{\!|\{\!|m|\!\}_b, a|\!\}_b\rangle, \langle -m\rangle\rangle, & \text{Strands} \\
\quad \emptyset, & \text{Node orderings} \\
\quad \{a^{-1}, b^{-1}\}, & \text{Non-origination} \\
\quad \{m\}) & \text{Unique origination}
\end{array}
$$

where the variable set that generates the algebra for the initiator role has been renamed so as to avoid conflicts with the variable set Z used by the preskeleton.

CPSA determines m is not kept secret by producing the shape in Figure 4. The added strands in the shape are instances of responder roles. The strands in the shape are:

$$
\begin{array}{l}
\langle +\{\!|\{\!|m|\!\}_b, a|\!\}_b\rangle \\
\langle -m\rangle \\
\langle -\{\!|\{\!|m|\!\}_b, a'|\!\}_b, +\{\!|\{\!|m|\!\}_{a'}, b|\!\}_{a'}\rangle \\
\langle -\{\!|\{\!|\{\!|m|\!\}_b, a|\!\}_b, a''|\!\}_b, +\{\!|\{\!|\{\!|m|\!\}_b, a|\!\}_{a''}, b|\!\}_{a''}\rangle
\end{array}
$$

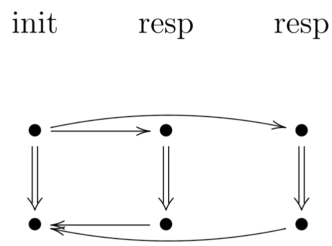

Figure 5: Exercise Skeleton

The non-origination and unique origination assumptions are as they are in the initial scenario preskeleton. An interesting exercise left for the reader is to produce a bundle that is realized by the shape.

4.2 Exercise

Consider the following roles.

$$\begin{aligned} \mathit{init}(a, b\colon A) &= \mathsf{r}(\langle +(a, b), -(b, a)\rangle, \emptyset, \emptyset) \\ \mathit{resp}(a, b\colon A) &= \mathsf{r}(\langle -(a, b), +(b, a)\rangle, \emptyset, \emptyset) \end{aligned}$$

Let $X = x, y\colon A$ and $k = \mathsf{k}_X(\langle \mathit{init}(a, b), \mathit{resp}(a, b), \mathit{resp}(a, b)\rangle,$
$\{\mathit{init}(a, b), \mathit{resp}(a, b)\},$
$\langle\langle +(x, y), -(y, x)\rangle,$
$\langle -(x, y), +(y, x)\rangle,$
$\langle -(x, y), +(y, x)\rangle\rangle,$
Node ordering in Figure 5,
$\emptyset,$
$\emptyset)$

What is $[\![k]\!]$?

One member is shown in Figure 6.

5 Penetrator Derivable Messages

To simplify notation, we write U_k to refer to U when $k = \mathsf{k}(rl, P, \Theta, \prec, N, U)$, and similarly for the other components of preskeleton k.

This section specifies what the penetrator can derive in a skeleton at a given reception node. In the section on the adversary model, it is explained

init	$\langle +(x, y), -(y, x)\rangle$
resp	$\langle -(x, y), +(y, x)\rangle$
resp	$\langle -(x, y), +(y, x)\rangle$
pair	$\langle -(y, x), -(y, x), +((y, x), (y, x))\rangle$
sep	$\langle -((y, x), (y, x)), +(y, x)\rangle$

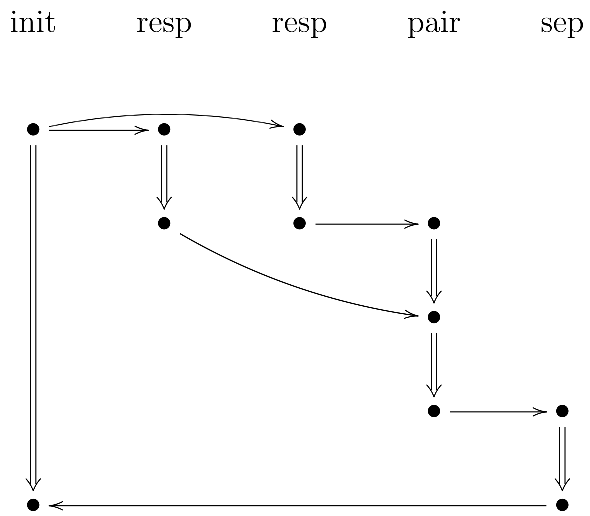

Figure 6: A Bundle Realized by the Example Skeleton

why the penetrator cannot use create roles for atoms in the what is called the exclusion set $\mathbf{X}_k = N_k \cup \{t \mid t \in U_k, |\mathcal{O}_k(t)| = 1\}$. At reception node n, the messages available to the penetrator due to message transmissions in the past are $\mathbf{T}_k(n) = \{t \mid n' \prec_k n, evt_k(n') = +t\}$. Therefore, for an algebra generated by X, the *public messages* available to the penetrator at node n are $\mathbf{P}_k(n) = \mathbf{T}_k(n) \cup (\mathfrak{B} \setminus \mathbf{X}_k) \cup X_{\text{MESG}} \cup \text{TAGS}$.

The penetrator roles derive messages.

$$\begin{aligned} D^0(T) &= T \\ D^{n+1}(T) &= \left\{ \begin{array}{r} (t_0, t_1) \mid t_0, t_1 \in D^n(T) \\ \{\!|t_0|\!\}_{t_1} \mid t_0, t_1 \in D^n(T) \\ t_0, t_1 \mid (t_0, t_1) \in D^n(T) \\ t_0 \mid \{\!|t_0|\!\}_{t_1}, inv(t_1) \in D^n(T) \end{array} \right\} \\ D(T) &= \textstyle\bigcup_{n\in\mathbb{N}} D^n(T) \end{aligned}$$

Here, $inv(t_1)$ is defined to be t_1^{-1} if t_1 : AKEY, and $inv(t_1)$ is otherwise defined to be t_1 so long as $t_1 \notin X_{\text{MESG}}$. A message t is derivable from T, written $T \vdash t$, if $t \in D(T)$. A message t is derivable at node n if $\mathbf{P}_k(n) \vdash t$.

Sometimes we may be interested in separating the notion of available messages from the notion of the *context*, which defines the set of derivable keys.

$$\begin{aligned} D^0(T, S) &= T \\ D^{n+1}(T, S) &= \left\{ \begin{array}{r} (t_0, t_1) \mid t_0, t_1 \in D^n(T, S) \\ \{\!|t_0|\!\}_{t_1} \mid t_0, t_1 \in D^n(T, S) \\ t_0, t_1 \mid (t_0, t_1) \in D^n(T, S) \\ t_0 \mid \{\!|t_0|\!\}_{t_1} \in D^n(T, S), S \vdash inv(t_1) \end{array} \right\} \\ D(T, S) &= \textstyle\bigcup_{n\in\mathbb{N}} D^n(T, S) \end{aligned}$$

In what follows, we find it useful to discuss the "minimum decryptions" available - that is, the messages we get by applying as much deconstruction as possible. We also are sometimes interested in this calculation when the set of messages available for deriving keys is distinct from the set of available messages. Let $\rightarrow$ be a reduction relation on pairs of sets of messages defined as follows:

$$\begin{aligned} (\{(t_0, t_1)\} \cup T, S) &\rightarrow (\{t_0, t_1\} \cup T, S) \\ (\{\{\!|t_0|\!\}_{t_1}\} \cup T, S) &\rightarrow (\{t_0, \{\!|t_0|\!\}_{t_1}\} \cup T, S) \\ &\rightarrow \text{if } t_1^{-1} \in D(S) \text{ and } t_0 \notin T \end{aligned}$$

The minimum decryption set $(M(T, S), S)$ is the normal form of relation $\rightarrow$, i.e. $(T, S) \rightarrow^* (M(T, S), S)$ and there is no (T', S') such that $(M(T, S), S) \rightarrow (T', S')$. Define $M(T)$ to be $M(T, T)$, and define $M(t, S)$ to be $M(\{t\}, S)$.

6 Authentication Tests

In a realized skeleton, the message at every reception node is derivable, but this is not so for an unrealized skeleton. A reception node that has a derivable message is called *realized*, and CPSA infers the existence of additional regular behavior by analyzing unrealized nodes.

It does so by identifying a so called critical message, a message carried by the node's message. The message is critical in the sense that the context in which it appears can only be explained by adding more regular strands, identifying messages, adding more constrains on node orderings, or various combinations of these actions.

Consider a reception node n that receives $\{\!|x|\!\}_{k_0}$, where critical message x is a uniquely originating symmetric key, and k_0 is an asymmetric key. In this case, x is being used as a nonce, and not for encryption, an artifact of algebra simplification. Assume that $+\{\!|x|\!\}_{k_1}$ is the only event that precedes n, where k_1^{-1} is a non-originating asymmetric key. Message $\{\!|x|\!\}_{k_0}$ is not derivable at n, because

$$\{\{\!|x|\!\}_{k_1}\} \cup (\mathfrak{B} \setminus \{x, k_1^{-1}\}) \cup X_{\mathrm{MESG}} \not\vdash \{\!|x|\!\}_{k_0}.$$

CPSA might explain this reception by identifying messages k_0 and k_1, or it might add a strand that receives $\{\!|x|\!\}_{k_1}$ and transmits x before node n if a role permits this new behavior.

A critical message might also be an encryption. Continuing the previous example, suppose that k_0 is non-originating, which makes $\{\!|x|\!\}_{k_0}$ into a critical message. CPSA might explain this reception by identifying messages k_0 and k_1, or it might add a strand that receives $\{\!|x|\!\}_{k_1}$ and transmits $\{\!|x|\!\}_{k_0}$ before node n if a role permits the new behavior.

We proceed with making the definition of a critical message precise by first considering the contexts of interest in which a critical message appears. For reception node n, the contexts are encryptions derived from the public messages at n, $\mathbf{P}(n)$, that contain the critical message. Furthermore, the encryptions are members of the minimum decryption set $M(\mathbf{P}(n))$ with underivable decryption keys. The context is called an escape set.

Definition 6.1 (Escape Set)**.** Let S and S' be sets of public messages. The *escape set* for t in messages S in context S' is $E(S,S',t) = \{\{\!|t_0|\!\}_{t_1} \in M(S,S') \mid t \sqsubseteq t_0 \wedge S \not\vdash t_1^{-1}\}$ when $t \notin M(S,S')$. Otherwise, $E(S,S',t) = \{t\}$.

We use the notation $E(S,t)$ as shorthand for $E(S,S,t)$; normally, the context is the set of messages.

The intuition is that, a message t_c that is carried by the message at n is critical when the contents of the escape set $E(\mathbf{P}_n(k), t_c)$ cannot be used to derive $mesg(n)$. To do so, the penetrator would have to decrypt a member of the escape set, which by definition it is not allowed to do. A critical message is one that has escaped the protection of previously transmitted encryptions, and CPSA infers more regular behavior in response.

We continue with the task of with making the definition of a critical message precise by stating what it means for an escape set to protect a message. Suppose t is carried by t', and S is a set of public messages. Furthermore, suppose that at every carried position at which t is carried in t', a subterm containing t is a member of the escape set $E(S,t)$. In this case, we say that term t is carried only within $E(S,t)$ in t', and observe that the subterm containing t is derivable because every member of the escape set is derivable. There is nothing about the fact that t' carries t that can be used to infer more regular behavior. An essential property of a critical message is that it is not carried only with the escape set in the message received at an unrealized node. The precise definition of carried only within follows.

Definition 6.2 (Ancestors)**.** For $t' = t \mathbin{@} p$, the *ancestors* of t' in t at p is the set $anc(t,p) = \{t \mathbin{@} p' \mid p' \text{ a prefix of } p\}$.

Definition 6.3 (Carried Only Within)**.** Term t is *carried only within* T in t', written $cow(t,T,t')$, if $p \in carpos(t,t')$ implies $anc(t',p) \cap T \neq \emptyset$. Term t *escapes* T in t', written $ncow(t,T,t')$, if $\neg(cow(t,T,t'))$, and therefore $ncow(t,T,t') = \exists p \in carpos(t,t')$ such that $anc(t',p) \cap T = \emptyset$.

Lemma 6.1. *If for every $u \in U$ we have that $cow(t_c,T,u)$, and we have that $cow(t_c,U,t')$ then $cow(t_c,T,t')$*

Proof. Let p is a carried position of t_c in t'. There is some ancestor $u_e \in anc(t',p)$ equivalent to a member of U. This ancestor u_e occurs at positions p' in t' where p' is a prefix of p. Let $p = p' \frown p''$; then since $cow(t_c,T,u_e)$ there is an ancestor $t_e \in anc(u_e,p'')$ equivalent to a member of T. But $t_e \in anc(t',p)$ so this occurrence of t_c is carried within T. □

Lemma 6.2. For any set of messages S, If $T_0 \subset T_1$ then for every t_c, for every $t \in E(S, T_0, t_c)$, $cow(t_c, E(S, T_1, t_c), t)$.

Proof. Let $t \in E(S, T_0, t_c)$ and let $p \in carpos(t_c, t)$. Note that $M(S, T_0) \subset D(S, T_1)$, since the enlarged context allows for possibly some more decryptions to be done, but all decryptions that can be done with the smaller context can still be done.

If t is an atom, it must be t_c, and therefore, $D(S, T_0) \vdash t_c$ so $D(S, T_1) \vdash t_c$, and t_c is a (non-proper) ancestor of itself.

Otherwise, $t = \{\!|t_0|\!\}_{t_1}$. Since $t \in E(S, T_0, t_c)$, $t \in M(S, T_0)$ and thus $t \in M(S, T_1)$. If $T_1 \not\vdash t_1^{-1}$ then $t \in E(S, T_1, t_c)$ and so p is carried within. Otherwise, one of two cases must happen: (1) $\exists t' = \{\!|t'_0|\!\}_{t'_1}$ in $anc(t, p)$ such that $T_1 \not\vdash {t'_1}^{-1}$ or (2) $t_c \in E(S, T_1, t_c)$. In the latter case, $t_c \in anc(t, p)$ so p is carried within. In the former case, assume t' is the largest such ancestor: then $t' \in E(S, T_1, t_c)$ and $t' \in anc(t, p)$, so p is carried within. □

In particular, the previous two lemmas imply that if $n' \prec n$ then for any set of messages S, and any t_c and any t', if $cow(t_c, E(S, \mathbf{P}_k(n'), t_c), t')$ then $cow(t_c, E(S, \mathbf{P}_k(n), t_c), t')$.

Lemma 6.3. Let S be a set of available messages and let t_c be a term such that either t_c is an atom or $t_c = \{\!|t_0|\!\}_{t_1}$ with $S \not\vdash t_1$. Then if $S \vdash t$ and $t_c \sqsubseteq t$, $cow(t_c, E(S, t_c), t)$.

Proof. If t is an atom, it cannot be derived from terms not carrying it. If t is an encryption, it can be derived from non-carrying terms only if its key is derivable.

Suppose $t \in D^n(S)$; we prove the theorem by induction. For $n = 0$, $D^0(S) = M(S)$. Suppose that $t_c \sqsubseteq_p t$. Then consider $anc(t, p)$—the encryptions on the path from t_c to t, including t_c. The minimal such encryption such that t_1^{-1} is not derivable from S will be in $E(S, t_c)$. Thus, any carried position of t_c within t is carried within $E(S, t_c)$.

Suppose $t \in D^n(S)$ but $t \notin D^{n-1}(S)$. Then either $t = (t_0, t_1)$ where $t_0, t_1 \in D^{n-1}(S)$, or $t = \{\!|t_0|\!\}_{t_1}$ where $t_0, t_1 \in D^{n-1}(S)$. In the former case, we must have that if $t_c \sqsubseteq_p t$ then either $p = 1 \frown p'$ and $t_c \sqsubseteq_{p'} t_0$, or $p = 2 \frown p'$ and $t_c \sqsubseteq_{p'} t_1$. In either case, there is some ancestor of p which is an ancestor of p' within t_0 or t_1, in $E(S, t_c)$ by inductive assumption. The case for $t = \{\!|t_0|\!\}_{t_1}$ is similar but since only the plaintext of an encryption is carried, all carried positions are of the form $1 \frown p'$ where $t_c \sqsubseteq_{p'} t_0$. □

Definition 6.4 (Target terms)**.** Let T be a set of terms, and let t_c be a term. Then the set of *target terms* containing t_c within T, denoted $\mathit{targ}(t_c, T)$ is the set $\{t | \exists t' \in T : t_c \sqsubseteq t \sqsubseteq t' \text{ but } t \notin T\} \cup \{t_c\}$.

A critical message may be either an atom or an encryption with an underivable encryption key. A critical message cannot be derived from its subterms. Here we define the notion of a critical position:

Definition 6.5 (Critical Position)**.** Position p is a *critical position* of t in the context of public messages S, written $p \in \mathit{critp}(S, t)$, iff

1. p is a carried position in t
2. $t \mathbin{@} p$ is an atom or $t \mathbin{@} p = \{\!|t_0|\!\}_{t_1}$ and $S \nvdash t_1$, and
3. $\mathit{anc}(t, p) \cap E(S, t \mathbin{@} p) = \emptyset$.

Theorem 6.1. $S \vdash t$ iff $\mathit{critp}(S, t) = \emptyset$.

A critical message is $t@p$ where p is a critical position. A critical message that is an atom is called a *nonce test*, and one that is an encryption is called an *encryption test*, and both types of tests are called an *authentication test*.

Definition 6.6 (Test Node)**.** Node n is a *test node* in k if $\mathit{evt}_k(n) = -t$ and $\mathit{critp}(\mathbf{P}_k(n), t) \neq \emptyset$.

CPSA makes progress by solving a test.

Definition 6.7 (Critical Position Solved)**.** Suppose p is a critical position at n in k, i.e. $\mathit{evt}_k(n) = -t$ and $p \in \mathit{critp}(\mathbf{P}_k(n), t)$, and suppose $k \stackrel{\phi,\sigma}{\longmapsto} k'$. Let $T = E(\mathbf{P}_k(n), t@p)$, $T' = \sigma(T)$, $n' = \phi(n)$, and $t' = \mathit{msg}_{k'}(n')$. Position p at n in k is *solved* in k', written $k \stackrel{n,p}{\twoheadrightarrow} k'$, if there exists a (ϕ, σ) such that:

1. $\mathit{anc}(t', p) \cap T' \neq \emptyset$, or
2. for some $t_p \in \mathbf{T}_{k'}(n')$, $\mathit{ncow}(t' \mathbin{@} p, T', t_p)$, or

2a. $\mathit{targ}(t'_c, T') \setminus \sigma(\mathit{targ}(t_c, T)) \neq \emptyset$ and there are variables in k's protocol that are not atoms, or

3. for some $\{\!|t_0|\!\}_{t_1} \in T'$, $\mathbf{P}_{k'}(n') \vdash t_1^{-1}$, or
4. $t' \mathbin{@} p = \{\!|t_0|\!\}_{t_1}$, and $\mathbf{P}_{k'}(n') \vdash t_1$.

In words, CPSA makes progress by a contraction (Item 1), where messages are identified, an augmentation (Item 2), where something is added to the escape set, or a listener augmentation (Item 3 and Item 4), where an assumption about the lack of the derivability of a key is shown to be invalid.

If solving a test is semantics preserving, and CPSA produces a finite set of skeletons that preserve the semantics at every step, CPSA will produce a set of realized skeletons that describe every possible bundle associated with an initial skeleton whenever CPSA terminates.

Theorem 6.2. For any skeleton k with an unrealized node n and a critical position p at n in k, $[\![k]\!] = \bigcup_{k' \in \{k' | k \xrightarrow{n,p}\!\!\!\!\rightarrow k'\}} [\![k']\!]$.

Proof. Let k be a skeleton in which n is an unrealized node, and t_c is a critical message at n in k. Let t be the message at n. Let k' be the skeleton of a bundle such that $k \stackrel{\phi,\sigma}{\longmapsto} k'$. Let $n' = \phi(n)$, let $t' = \sigma(t)$. Let $T = E(\mathbf{P}_k(n), t \mathbin{@} p)$, and let $T' = \sigma(T)$. Let $S' = \mathbf{P}_{k'}(n')$.

Let $t_c = t \mathbin{@} p$ and $t'_c = t' \mathbin{@} p$. Because k' is the skeleton of a bundle, there is no critical message at n'. Therefore, t'_c is not a critical message at n' in k'. That is, there is no position p' such that $t' \mathbin{@} p' = t'_c$ and p' is a critical position at n' in k'.

If $t'_c = \{\!|t_0|\!\}_{t_1}$ and $S' \vdash t_1$ then by condition 4 of the solved definition, $k \xrightarrow{n,p}\!\!\!\!\rightarrow k'$.

Otherwise, $cow(t'_c, E(S', t'_c), t')$.

Suppose that $\forall t_e \in E(S', t'_c)$, $cow(t'_c, T', t_e)$. Since we know $cow(t'_c, E(S', t'_c), t')$, by Lemma 6.1, $cow(t'_c, T', t')$. Thus, since $t' \mathbin{@} p = t'_c$, $anc(t', p) \cap T' \neq \emptyset$ and thus $k \xrightarrow{n,t_c}\!\!\!\!\rightarrow k'$ by condition (1) of the definition of solved.

Otherwise, there is some $t_e \in E(S', t'_c)$ such that $ncow(t'_c, T', t_e)$. If t_e is not an encryption, it must be that $E(S', t'_c) = \{t'_c\}$ and that t'_c is an atom. In this case, note that $t'_c \in \mathbf{X}_{k'}$ because $t_c \in \mathbf{X}_k$ and because (ϕ, σ) is a homomorphism. Thus, regardless of whether t_e is an encryption or not, $(\mathfrak{B} \setminus \mathbf{X}_k) \cup X_{\mathrm{MESG}} \nvdash t_e$, but since $t_e \in E(S', t'_c)$, we know that $t_e \in M(S')$. Therefore, t_e can be derived from some public message.

To make this precise, define $M_0(t_p, S')$ recursively as follows:

- $t_p \in M_0(t_p, S')$.
- If $\{\!|t_0|\!\}_{t_1} \in M_0(t_p, S')$ and $S' \vdash t_1^{-1}$ then $t_0 \in M_0(t_p, S')$.
- If $(t_0, t_1) \in M_0(t_p, S')$ then $t_0, t_1 \in M_0(t_p, S')$.

Then define $M(t_p, S')$ to be the all the non-pairs in $M_0(t_p, S')$.

In other words, $M(t_p, S')$ is the portion of $M(S')$ derivable from t_p using keys derivable from S'. It is clear that $M(S') = M((\mathfrak{B}\backslash\mathbf{X}_k)\cup X_{\text{MESG}})\cup_{t_p\in\mathbf{T}_{k'}(n')} M(t_p, S')$. So let t_p be such that $t_e \in M(t_p, S')$.

Define q to be a position such that $t_p@q = t_e$ and such that for every proper prefix q'' of q, either $t_p@q''$ is a pair, or $t_p@q'' = \{\!|t_0|\!\}_{t_1}$ where $S' \vdash t_1^{-1}$ and where $q'' \frown 1$ is also a prefix of q. In other words, let q be a position at which t_e is carried in t_p *and* derivable. We know such a q must exist because $t_e \in M(t_p, S')$.

Since $ncow(t'_c, T', t_e)$, let q' be a carried position of t'_c within t_e such that no ancestor is in T'. Consider position $q \frown q'$. If there is some position $q \frown q''$ for q'' a prefix of q' such that $t_p \mathbin{@} q \frown q''$ is in T' then the same could be said of $t_e \mathbin{@} q''$, but this would be a contradiction. So either there is no ancestor in $anc(t_p, q \frown q')$ equivalent to a member of T' (in which case $k \xrightarrow{n,p} k'$ by condition (2) of the definition of solved), or there is some position q'' such that $t_p \mathbin{@} q''$ is equivalent to some $u \in T'$. By our choice of q, and by the fact that any such u must necessarily be an encryption[1], it follows that $u = \{\!|t_0|\!\}_{t_1}$ where $S' \vdash t_1^{-1}$. In this case, $k \xrightarrow{n,p} k'$ by condition (3) of the definition of solved.

Thus allows us to conclude that for every bundle Υ denoted by k, there is a skeleton k', namely, the skeleton of Υ, such that $k \xrightarrow{n,t_c} k'$. Since Υ is denoted by k', this proves that $[\![k]\!] \subseteq \bigcup_{k'\in\{k' | k \xrightarrow{n,t_c} k'\}} [\![k']\!]$. The other direction is far simpler: we just note that for each k' such that $k \xrightarrow{n,t_c} k'$, there is a homomorphism from k to k', so the set of bundles denoted by k' is a subset of those denoted by k. This completes the proof.

□

7 Test Solving Algorithm

This section describes the algorithm undertaken by CPSA in order to find realized skeletons that include the structural assumptions of the "point of view," the initial input.

[1] The only case in which a value in T' is not an encryption is when $t_c \in M(S)$ and t_c is an atom, which we know is false here.

7.1 Primitive Preskeleton Operators

The are four primitive operators on preskeletons used by CPSA to solve authentication tests. Each operator is a partial map from preskeletons to preskeletons.

Definition 7.1 (Substitution Operator)**.** For order-sorted substitution $\sigma\colon X \to \mathfrak{A}_Y$, the operator $\mathbb{S}_\sigma$ is:

$$\begin{array}{l}\mathbb{S}_\sigma(\mathsf{k}_X(rl, P, \Theta_X, \prec, N, U)) = \\ \quad \mathsf{k}_Y(rl, P, s \mapsto \sigma \circ \Theta_X(s), \prec, \sigma(N), \sigma(U))\end{array}$$

For $k' = \mathbb{S}_\sigma(k)$, there is a homomorphism from k to k' only if for all $t \in U_k$, $\mathcal{O}_k(t) \subseteq \mathcal{O}_{k'}(\sigma(t))$. The structure preserving maps associated with the homomorphism are ϕ_{id} and σ.

Definition 7.2 (Compression Operator)**.** For distinct strands s and s', operator $\mathbb{C}_{s,s'}$ compresses strand s into s'.

$$\begin{array}{l}\mathbb{C}_{s,s'}(\mathsf{k}_X(rl, P, \Theta_X, \prec, N, U)) = \\ \quad \mathsf{k}_X(rl \circ \phi'_s, P, \Theta_X \circ \phi'_s, \prec', N, U)\end{array}$$

where

$$\phi'_s(j) = \begin{cases} j+1 & \text{if } j \geq s \\ j & \text{otherwise,} \end{cases}$$

relation $\prec'$ is the transitive closure of $\phi_{s,s'}(\prec)$, and

$$\begin{aligned}\phi_{s,s'}(j) &= \begin{cases} \phi_s(s') & \text{if } j = s \\ \phi_s(j) & \text{otherwise} \end{cases} \\ \phi_s(j) &= \begin{cases} j-1 & \text{if } j > s \\ j & \text{otherwise.} \end{cases}\end{aligned}$$

The compression operator is only used when $\Theta_X(s)$ is a prefix of $\Theta_X(s')$, and when there is a homomorphism from k to $\mathbb{C}_{s,s'}(k)$. The structure preserving maps associated with the homomorphism are $\phi_{s,s'}$ and σ_{id}. Note that the compression operator is defined only when relation $\prec'$ is asymmetric, and that $\phi_{s,s'} \circ \phi'_s = \phi_{\mathrm{id}}$.

Definition 7.3 (Ordering Enrichment Operator)**.** Operator $\mathbb{E}(k)$ enriches $\prec_k$ by adding all elements implied by unique origination.

The ordering enrichment operator is total and idempotent. The structure preserving maps associated with the operator's homomorphism are ϕ_{id} and σ_{id}, i.e. the homomorphism is an embedding.

Definition 7.4 (Augmentation Operator)**.** For node n, role r, and trace C, operator $\mathbb{A}_{n,r,C}$ is:

$$\begin{array}{l}\mathbb{A}_{n,r,C}(\mathsf{k}_X(rl, P, \Theta_X, \prec, N, U)) = \\ \quad \mathsf{k}_{X'}(rl \frown r, P, \Theta_X(s) \frown C, \prec', N', U')\end{array}$$

where X' is X extended to include the variables in C, $\prec'$ is the minimal extension of $\prec$ such that $(|\Theta_X| + 1, |C|) \prec' n$, N' is N extended with non-origination assumptions inherited from r by C, and likewise for U'.

The structure preserving maps associated with the augmentation operator's homomorphism are ϕ_{id} and σ_{id}, i.e. the homomorphism is an embedding.

7.2 Test solving steps

Suppose k is a skeleton with a critical position p at node n. Let $-t = evt_k(n)$, $t_c = t \,@\, p$, and $T_e = E(\mathbf{P}_k(n), t_c)$. Pre-skeletons k' produced by the following steps make up the "pre-cohort" $\mathcal{PC}_{k,n,p}$ of k with respect to the test (n, p).

Contraction: $k' = \mathbb{S}_\sigma(k)$, where σ is a most general unifier such that for some $t_a \in anc(t, p)$ and $t_e \in T_e$, $\sigma(t_a) = \sigma(t_e)$.

Augmentation: $k' = \mathbb{A}_{n,r,C}(\mathbb{S}_\sigma(k))$, where n, r, C, and σ are as described in Section 7.5.

Displacement: $k' = \mathbb{C}_{s,s'}(\mathbb{A}_{n,r,C}(\mathbb{S}_\sigma(k)))$, where n, r, C, and σ are as described in Section 7.5, where s and s' are the newly created strand and any other existing strand (where s' is the strand with greater height, if the heights are unequal).

Escape set listeners: For $t_e \in T_e$, if $t_e = \{\!|t_0|\!\}_{t_1}$ and $C = \langle -t_1^{-1}, +t_1^{-1} \rangle$ then $k' = \mathbb{A}_{n,lsn,C}(k)$.

Critical message listener: If $t_c = \{\!|t_0|\!\}_{t_1}$ then $k' = \mathbb{A}_{n,lsn,\langle -t_1, +t_1 \rangle}(k)$.

By definition, when $k \xrightarrow{n,p} k'$, there is a homomorphism $k \xmapsto{\phi,\sigma} k'$, where maps ϕ and σ are the composition of the maps from the steps used to perform a test solving reduction. Since each operator does not ensure that the node at which each uniquely originating atom originates is preserved, skeletons that do not meet this requirement must be filter out. To perform the filtering, the implementation computes σ and ϕ.

Conjecture 7.1 (Authentication Solving Algorithm Complete)**.** Suppose k is a skeleton with a critical position p at node n, and p at n in k is solved in skeleton k', i.e. $k \overset{n,p}{\twoheadrightarrow} k'$. Then there exists a skeleton k'', strand map ϕ, and substitution σ such that $k \xrightarrow{n,p} k''$, and $k'' \xmapsto{\phi,\sigma} k'$.

The proof appears to be too hard. Instead we focus on the following conjecture.

Definition 7.5 (Listener expanded bundle)**.** Let bundle Υ be a run of protocol. Its *listener expanded bundle* is $\mathit{lsn}(\Upsilon)$, which is Υ after inserting a listener after every message transmitted by a non-listener strand.

Conjecture 7.2. Suppose k is a skeleton with a critical position p at node n. For all $\Upsilon \in [\![k]\!]$ and the k' that realizes $\mathit{lsn}(\Upsilon)$, there exists a skeleton k'', strand map ϕ, and substitution σ such that $k \xrightarrow{n,p} k''$, and $k'' \xmapsto{\phi,\sigma} k'$.

7.3 Hulling process

A preskeleton differs from a skeleton only in that in a skeleton, terms assumed to be uniquely originating originate at no more than one node, and that receptions of a uniquely originating term that originates need not be ordered after its unique point of origination.

The process of hulling takes as input a quadruple (k_0, k, ϕ, σ) where k_0 is a skeleton, k is a preskeleton, and $k_0 \xmapsto{\phi,\sigma} k$, and outputs a set $\mathbb{H}(k_0, k, \phi, \sigma)$ of skeletons k' such that for each k' there is a ϕ', σ' such that $k_0 \xmapsto{\phi'\circ\phi,\sigma'\circ\sigma} k'$, and such that for any k'' such that $k_0 \xmapsto{\phi''\circ\phi,\sigma''\circ\sigma} k''$, this map factors through one of the skeletons and through its associated map in $\mathbb{H}(k_0, k, \phi, \sigma)$.

$\mathbb{H}(k_0, k, \phi, \sigma)$ is produced from k by iteratively resolving every instance of a uniquely originating term originating at more than one node, and then applying order enrichment. A case of multi-origination can be resolved in one of two ways. If t originates on distinct strands s and s' where the height of s' is no less than the height of s, then $k' = \mathbb{C}_{s,s'}(\mathbb{S}_\sigma(k))$ where σ is a most general unifier of $\Theta_k(s)$ and $\Theta_k(s')$ is a *normal hulling*. If t originates on distinct strands s and s' and s' is not in the image of k_0, then $k' = \mathbb{S}_\sigma(k)$ where σ is a most general algebra homomorphism such that $\sigma \circ \Theta_k(s')$ does not originate $\sigma(t)$ (for instance, due to $\sigma(t)$ being received first), then k' is a *de-origination.*

Theorem 7.1. The process of computing $\mathbb{H}(k_0, k, \phi, \sigma)$ produces a hulling as defined above.

No proof yet.

7.4 Pruning process

Let a *subskeleton* of a skeleton be defined as a subset of the strands. Consider a skeleton k along with a "point of view" skeleton k_0 with $k_0 \overset{\phi_0,\sigma_0}{\longmapsto} k$.

Two subskeletons S_0, S_1 are *essentially identical* if:

- $|S_0| = |S_1$ and $S_0 \cap S_1 = \emptyset$
- $(S_0 \cup S_1) \cap \phi_0(k_0) = \emptyset$.
- There exists a bijection ϕ between S_0 and S_1 such that for each $s \in S_0$, $\phi(s)$ is an instance of the same role as s and has the same length as s.
- There is an algebra homomorphism σ that is a renaming such that $\sigma(evt(s,i)) = evt(\phi(s), i)$.
- For all variables v appearing in $k \setminus (S_0 \cup S_1)$, σ is the identity on v.
- For any pair of nodes $(s,i), (s',i')$ in the first subskeleton, $(s,i) \prec (s',i')$ if and only if $(\phi(s), i) \prec (\phi(s'), i')$.
- For any pair of nodes $(s,i), (s',i')$ where (s,i) is in the first subskeleton and (s',i') is in neither subskeleton, $(s,i) \prec (s',i') \Leftrightarrow (\phi(s), i) \prec (s',i')$ and $(s',i') \prec (s,i) \Leftrightarrow (s',i') \prec (\phi(s), i)$.

CPSA "prunes" away essentially identical strands; in future versions, CPSA may be capable of pruning identical multi-strand subskeletons.

When a skeleton k has two essentially identical subskeletons S_0, S_1, there is a well-defined $\mathbb{P}_{S_0,S_1}(k)$ in which all of S_1 is simply removed from k. Under some circumstances, there may be a homomorphism from k to $\mathbb{P}_{S_0,S_1}(k)$ but in others there may not. However, there will always be a homomorphism from k_0 to $\mathbb{P}_{S_0,S_1}$.

7.5 Augmentation

An augmentation step is used to add a strand to a skeleton. An augmentation step has the form $\mathbb{A}_{n,r,C} \circ \mathbb{S}_\sigma$. Suppose skeleton $k = \mathsf{k}_X(\_, P, \_, \_, \_, \_)$ has a critical position p at node n, and $t_c = \mathit{msg}_k(n) \mathbin{@} p$ is the critical message. CPSA computes the parameters for a set of augmentation steps as follows. First, compute the target messages. Let $T_e = E(\mathbf{P}_k(n), t_c)$. The target messages are $\mathit{targ}(t_c, T_e)$. Next, for each $\mathsf{r}_Y(C, N, U) \in P$ and each index h where $C(h) = +t$, a transmission, do the following.

Create fresh variables: Let σ_r be a sort preserving variable renaming, where the domain is the variables that occur in $C|_h$, and every variable in the range does not occur in X or $\mathit{Vars}(P)$.

Insert critical message: For each message t' carried by t, and each $t_t \in \mathit{targ}(t_c, T_e)$, consider most general unifiers σ' where, $\sigma'(t') = \sigma'(t_t)$ and $\sigma_r \trianglelefteq \sigma'$.

Ensure previous events do not transform: For each σ', find most general unifiers σ such that for $1 \leq i < h$, $\mathit{cow}(\sigma(t_c), \sigma(T_e), \sigma(C(i)))$ and $\sigma' \trianglelefteq \sigma$. Let $S_{r,h}$ be a set of substitutions σ with non-most general unifiers removed.

Ensure last event transforms: For each $\sigma \in S_{r,h}$, if $\mathit{ncow}(\sigma(t_c), \sigma(T_e), \sigma(C(h)))$, try augmenting with parameters n, r, $\sigma \circ C|_h$, and σ.

7.6 Preskeleton Reductions

Function *skel* is a partial function that maps preskeletons to sets of pruned skeletons. If given a preskeleton k where some uniquely originating atoms originate more than once, *skel* applies hulling steps so as to eliminate uniquely originating atoms that originate more than once or is undefined. Otherwise, it applies the ordering enrichment operator once to produce a skeleton. Finally, it applies as many pruning steps as is possible to produce a skeleton that is pruned subject to restriction that only pairs of strands are considered.

As remarked above, it may not always be the case that $k \stackrel{\phi,\sigma}{\longmapsto} k'$ where $k' \in \mathit{skel}(k)$.

8 Completeness of CPSA

NOTE: This section is still an unstable draft!

In this section, we prove that the CPSA cohort of finding regular and listener augmentations and contractions to solve a test is complete in the sense that if k is a skeleton and $k \stackrel{\phi,\sigma}{\longmapsto} k'$ and $k \stackrel{n,p}{\twoheadrightarrow} k'$ then there is some k'' directly obtainable from k and a factorization $k \stackrel{\phi_0,\sigma_0}{\longmapsto} k'' \stackrel{\phi_1,\sigma_1}{\longmapsto} k'$ such that $\phi = \phi_1 \circ \phi_0$ and $\sigma = \sigma_1 \circ \sigma_0$.

This would be the ideal theorem statement but we need to make the statement more complicated because of CPSA's use of listener augmentations which are not reflected in bundles. So instead, we will have to prove that if k is a skeleton that denotes a bundle Υ, then there is some k'' directly obtainable from k that denotes Υ. In order to relate this to the solved predicate, we make use of the theorem from the previous section, which states that if $k \stackrel{\phi,\sigma}{\longmapsto} k'$ and k' is realized, then $k \stackrel{n,p}{\twoheadrightarrow} k'$ for any unrealized n in k with critical position p. Let k' be the skeleton of bundle Υ. For the non-listener cases we will prove the homomorphism from k to k' factors through k''; for the listener cases, we will simply prove that Υ is still denoted by k''.

Let k be an unrealized skeleton and let n be an unrealized node and p be a critical position at that node. Let $\Upsilon(\Theta, \rightarrow)$ be a bundle denoted by k and let k' be the skeleton of that bundle. Note that $k \stackrel{\phi,\sigma}{\longmapsto} k'$ and $k \stackrel{n,p}{\twoheadrightarrow} k'$. The proof proceeds by cases, corresponding to the conditions of the definition of solved.

[Condition (1) of solved]: If p' is a prefix of p such that $t' \mathbin{@} p' = t'_e \in T'$, then σ is a unifier of $t \mathbin{@} p'$ and t_e where $\sigma(t_e) = t'_e$. Specifically, let k'' be nodewise isomorphic to k, and let σ_0 be a most general unifier of t_e and $t \mathbin{@} p'$ more general than σ so that $\sigma = \sigma_1 \circ \sigma_0$. Then let ϕ_0 be the nodewise isomorphism and let $\phi_1 = \phi \circ \phi_0^{-1}$. We have now specified $\sigma_0, \sigma_1, \phi_0, \phi_1$, and note that $\phi = \phi \circ \phi_0^{-1} \circ \phi_0$ and that $\sigma = \sigma_1 \circ \sigma_0$, which proves the factorization commutes.

[Conditions (3) and (4) of solved]: If $\mathbf{P}_{k'}(n') \vdash {t'_1}^{-1}$ where $\{\!|t'_0|\!\}_{t'_1} \in T'$, a listener augmentation will serve as our intermediate step. CPSA attempts a listener augmentation for every decryption key associated with an encryption in its escape set; let $\{|t_0|\}_{t_1}$ be (one) preimage of $\{|t'_0|\}_{t'_1}$ under σ in the escape set T. CPSA will thus attempt a listener augmentation for t_1^{-1}. Let k'' be defined to be k with an additional listener for t_1^{-1}. The map ϕ_0 is nodewise injective with image that avoids the new listener, and the map σ_0 is the identity. Let k'_1 be k' with an additional listener for ${t'}_1^{-1}$, ordered immediately before n' (that is, the listener nodes are previous to n' and all nodes of k' strictly previous to n' are strictly previous to the first node of the listener.) We can define a homomorphism from k'' to k'_1, namely, ϕ_1, σ_1 where $\sigma_1 = \sigma$ and where ϕ_1 is defined to be $\phi \circ \phi_0^{-1}$ for nodes in the image of ϕ_0, with the two listener nodes mapping to the two listener nodes of k'_1. k'_1 will still be realized; the only new reception is for ${t'}_1^{-1}$ but we assumed that $\mathbf{P}_{k'}(n') \vdash {t'}_1^{-1}$ and all public messages sent before n' are available before the listener reception since n' contains a reception event.

In $\Upsilon(\Theta, \rightarrow)$, we must now insert the listener in a way compatible with k'_1, yet also keep the bundle property intact. This may require one or more penetrator strands be added to produce ${t'}_1^{-1}$ if it was not already produced. Let $\Theta', \rightarrow'$ be the extensions of $\Theta, \rightarrow$ to incorporate the listener strand and any necessary penetrator strands. Then k'_1 realizes $\Upsilon(\Theta', \rightarrow')$ and so k'' realizes $\Upsilon(\Theta, \rightarrow)$ modulo listeners.

If $t'_c = \{|t'_0|\}_{t'_1}$ and $\mathbf{P}_{k'}(n') \vdash t'_1$ then, again, we use a listener augmentation. When the critical term is an encryption, CPSA will attempt a listener augmentation for its encryption key. In this case, if $t_c = \{|t_0|\}_{t_1}$, CPSA will attempt a listener augmentation for t_1. The rest of the argument for this case is effectively identical to the argument for condition (3), but the listener in k'' will be for t_1 rather than t_1^{-1} and the listener in k'_1 will be for t'_1 rather than ${t'}_1^{-1}$.

[Condition (2) of solved]: The proof of this case proceeds via the proof of two lemmas. The first lemma proves that a good candidate augmentation exists, and the second lemma proves that this augmentation is successful, produces a distinct result from k, and covers k'.

Definition 8.1. Let $r = (C, N, U)$ be a role, i be whole number, pp be a position, and tt be a term. Then (r, i, pp, tt) is an *augmentation candidate* for (k, n, p) if:

1. $|C| \leq i$,
2. Position pp in $C(i)$ is a well-defined, carried position.

3. tt is either $t@p$ or a proper, carried subterm of an element of $E(\mathbf{P}_n(k), t@p)$ that carries $t @ p$.

Definition 8.2. Let k be an unrealized skeleton with unrealized node n and critical position p. Let k' be a realized skeleton, and let $k \overset{\phi,\sigma}{\longmapsto} k'$. Then (r, i, pp, tt) is a *solving augmentation candidate* for $(k, k', n, p, \phi, \sigma)$ if (r, i, pp, tt) is an augmentation candidate for (k, n, p), and

1. There is a send of a message t' at a node $n' = (s, i)$ in k' such that (1) $n' \prec \phi(n)$, (2) strand $\Theta'(s)$ is an instance of role r, (3) $t' @ pp = \sigma(tt)$, and (4) if $t'' = msg_k(s, j)$ for $j < i$ then $cow(t'_c, T', t'')$.

2. There is some position q such that $t'_c \sqsubseteq_q t'$ and such that (1) $anc(t', q) \cap T' = \emptyset$ or (2) there exists some $q', q'' : q' \frown q''$ is a prefix of q and $t' @ q' \in T'$ and $t' @ q' \frown q'' \in targ(t'_c, T')$ but $t' @ q' \frown q'' \notin \sigma(targ(t_c, T))$.

Lemma 8.1. If $k \overset{n,p}{\twoheadrightarrow} k'$ via $k \overset{\phi,\sigma}{\longmapsto}$ and only condition (2) of the solved predicate holds, then there exists a solving augmentation candidate for $(k, k', n, p, \phi, \sigma)$.

Proof. In order to concisely deal with similarities between more complex cases and simpler cases, we define several related sequences which become defined as needed. Specifically, we define a sequence T'_i of sets of terms, a sequence t'_i of terms, sequences p_i and p'_i of positions, and a sequence n_i of nodes, such that:

- $T'_i \subset T'$.
- For $i > 0$, $T'_i = \{t'_i\} \cup (E(\mathbf{P}_{k'}(n_i), \mathbf{P}_{k'}(n'), t'_c) \cap T'_{i-1})$.
- For $i > 0$,p'_i is a prefix of p_i.
- For $i > 0$, $t'_c \sqsubseteq_{p_i} t'_i$ but $t'_i @ p'_i \in targ(t'_c, T')$ and $t'_i @ p'_i \notin \sigma(targ(t_c, T))$
- For $i \geq 0$, $n_{i+1} \preceq n_i$, and $n_0 \preceq n'$.
- $ncow(t'_c, T'_i, msg(n_i))$.

We let $T'_0 = T'$ and we let n_0 be the node preceding n' transmitting the message t_p such that $ncow(t'_c, T', t_p)$. There is no need to define t'_0, p_0, or p'_0 because no conditions are placed on them.

Note that in fact $T'_i = \{t'_i\} \cup (E(\mathbf{P}_{k'}(n_i), \mathbf{P}_{k'}(n'), t'_c) \cap T')$, by repeated substitution of T'_{i-1} coupled with the fact that these escape sets get smaller as i grows.

Since T' is finite, we know that this sequence cannot be defined farther than $i = |T'|$. Therefore, it suffices to prove that either we can find a solving augmentation candidate given a sequence of length i, or that we can extend the sequence while maintaining these properties.

Recall that since for all $\{\!|t_0|\!\}_{t_1} \in T'$ we have that $\mathbf{P}_{k'}(n') \not\vdash t_1^{-1}$, and also if t'_c is an encryption, its encryption key is not derivable from $\mathbf{P}_{k'}(n')$, we know that if S is a set of messages such that for every $x \in S$ we have that $cow(t'_c, T'_i, x)$, then $S \vdash x'$ implies that $cow(t'_c, T', x')$. Thus, for any reception node prior to n' in k', if the reception carries t'_c not only within T'_i then there is an earlier send node that carries t'_c not only within T'_i; if the earliest such node is a reception, its message would not be derivable.

Thus, for the message at any node $\preceq n_i$ in k', if that message does not carry t'_c only within T'_i, then either that node is a send, or there is an earlier node whose message does not carry t'_c only within T'. In other words, if such a message exists, we can find an ealiest transmission node ν_i sending τ_i such that $ncow(t'_c, T', \tau_i)$ and for all τ sent earlier than ν_i, $cow(t'_c, T', \tau)$. Note that since $ncow(t'_c, T'_i, msg(n_i))$, there must exist some node ν_i such that (1) $evt_{k'}(\nu_i) = +\tau_i$, (2) $ncow(t'_c, T'_i, \tau_i)$, and (3) for all $\nu \prec \nu_i$, $cow(t'_c, T'_i, msg(\nu))$. In other words, ν_i is an earliest transmission node whose message carries t'_c not only within T'_i.

Since k' is realized, we know that $cow(t'_c, E(\mathbf{P}_{k'}(\nu_i), t'_c), \tau_i)$ and therefore, $cow(t'_c, E(\mathbf{P}_{k'}(n_i), \mathbf{P}_{k'}(n'), t'_c), \tau_i)$ by Lemma 6.2. Since $ncow(t'_c, T'_i, \tau_i)$, let q be such that $t'_c \sqsubseteq_q \tau_i$ and $anc(\tau_i, q) \cap T'_i = \emptyset$. If such a q exists such that $anc(\tau_i, q) \cap T' = \emptyset$ then let q have this property. Let $t^\star \in anc(\tau_i, q) \cap E(\mathbf{P}_{k'}(n_i), \mathbf{P}_{k'}(n'), t'_c) \setminus T'_i$, where $t^\star \in T'$ if such a $t^\star$ exists given our choice of q. If $t^\star \in T'$ then $t^\star = t_i$ (recall, $T'_i \setminus (T' \cap E(\mathbf{P}_{k'}(n_i), \mathbf{P}_{k'}(n'), t'_c)) = \{t'_i\}$). In such a case, assume without loss of generality that we can write $q = q' \frown p_i$ where $\tau_i \mathbin{@} q' = t^\star$.[2]

Note that q satisfies condition (3) of a solving augmentation candidate, a condition which depends only on the transforming message, τ_i. If $t^\star \notin T'$ then it is because $anc(\tau_i, q) \cap T' = \emptyset$. Otherwise, $t^\star = t_i$, and $q \frown p'_i$ is the prefix required: $\tau_i \mathbin{@} q' = t_i \in T'$ and $\tau_i \mathbin{@} q' \frown p'_i = t_i \mathbin{@} p'_i$ is in $targ(t'_c, T')$ but

[2] If not, write $q = q_0 \frown q_1$ where $t' \mathbin{@} q_0 = t^\star$ and $t'_c \sqsubseteq_{q_1} t^\star$. Consider $q' = q_0 \frown p_i$. Then $t'_c \sqsubseteq_{q'} \tau_i$ and $t^\star \in anc(\tau_i, q')$.

not in $\sigma(targ(t_c, T))$.

From here, we must identify our solving augmentation candidate or identify our extension of the sequence.

Let $r_i = (C_i, N_i, U_i)$ be the role associated with ν_i's strand in k' via its role map, and let $\nu_i = (s_i, h_i)$. We know that ν_i is a sending node, so let t_r be such that $C(h_i) = +t_r$. If q is a well-defined position of t_r then (r_i, h_i, q, t_c) is a solving augmentation candidate. (Proof omitted.)

If not, then let q' be the longest prefix of q such that q' is a well-defined position of t_r (and let $q = q' \frown q''$); then $t_r \mathbin{@} q' = x$ where x is of sort MESG. Since our protocol satisfies the acquired constraint there is an earlier message $C(h_{aq}) = -t_{aq}$ in which x is acquired; let p_{aq} be such that $x \sqsubseteq_{p_{aq}} t_{aq}$. If $t'_{aq} = msg(n'_{aq})$ where $n'_{aq} = (s_i, h_{aq})$ then we know that $t'_{aq} \mathbin{@} (p_{aq} \frown q'') = t'_c$ so there must be some ancestor t'_e in $T'_i \cap anc(t'_{aq}, p_{aq} \frown q'')$. Specifically, let t'_e be the ancestor closest to $t'_{aq} \mathbin{@} (p_{aq} \frown q'')$ in $E(\mathbf{P}_{k'}(n'_{aq}), \mathbf{P}_{k'}(n'), t'_c)$. We know there is some ancestor in $E(\mathbf{P}_{k'}(n'_{aq}), t'_c)$ because k' is realized, and by Lemma 6.2 and Lemma 6.1, there is some ancestor in $E(\mathbf{P}_{k'}(n'_{aq}), \mathbf{P}_{k'}(n'), t'_c)$. Given that there is at least one, we can without loss of generality pick the smallest as t'_e.

Furthermore, t'_e must appear at some position p'_{aq} in t'_{aq} which is a proper prefix of p_{aq}, otherwise, t'_e would be in $anc(t_p, q)$. Let $p_{aq} = p'_{aq} \frown p''_{aq}$. If there is some target term tt such that $\sigma(tt) = t'_e \mathbin{@} p''_{aq}$ then (r_i, h_i, q', tt) is a solving augmentation candidate. (Proof omitted.)

Otherwise, for all $tt \in targ(t_c, T)$, $\sigma(tt) \neq t'_e \mathbin{@} p''_{aq}$. Let $t'_{i+1} = t'_e$, let $p_{i+1} = p''_{aq}$, let $T'_{i+1} = (T'_i \setminus \{t'_e\}) \cap E(\mathbf{P}_{k'}(n'_{aq}), \mathbf{P}_{k'}(n'), t'_c)$, and let n_{i+1} be some node $\preceq n'_{aq}$ such that $t'_e \in M(mesg(n_{i+1}), \mathbf{P}_{k'}(n'))$; we know such a node exists because $t'_e \in E(\mathbf{P}_{k'}(n'_{aq}), \mathbf{P}_{k'}(n'), t'_c)$. Then $ncow(t'_c, T'_{i+1}, mesg(n_{i+1})$ because $t'_e \in M(mesg(n_{i+1})$ and t'_e has at least one carried position of t'_c, namely p''_{aq}, that has no ancestor in $E(\mathbf{P}_{k'}(n'_{aq}), \mathbf{P}_{k'}(n'), t'_c)$. □

This establishes that a solving augmentation candidate exists. Next we must prove that the existence of such an augmentation candidate implies a cohort member that we can factor through.

Lemma 8.2. If (r, i, pp, tt) is a solving augmentation candidate for $(k, k', n, p, \phi, \sigma)$ then if $k'' = aug(k, r, i, pp, tt)$ then there exist homomorphisms (ϕ', σ') and (ϕ'', σ'') such that $k \overset{\phi',\sigma'}{\longmapsto} k'' \overset{\phi'',\sigma''}{\longmapsto} k'$ where $\phi = \phi'' \circ \phi'$ and $\sigma = \sigma'' \circ \sigma'$, and k'' is not isomorphic to k.

STOP READING HERE

Proof. In such cases, the map from k to k' filters through k'' which is the result of an augmentation followed by pruning. To specify an augmentation operation, we must specify an n which the new instance precedes, a role r, a trace C (an instantiation of r under some substitution up to some height) and a substitution σ to apply in conjunection with the augmentation. This produces a pre-skeleton k''_0 and then k'' is the result of first hulling k''_0 and then pruning single strands.

n' will serve as the n for the augmentation, and r will be the role.

In k', there is a node $\nu = (s', i)$ such that $\nu \prec n'$ such that $evt(\nu) = +\tau$ and $\tau \mathbin{@} pp = \sigma(tt)$, and where s is an instance of role r, and such that for all $j < i$, $cow(t'_c, T', msg_k(s, j))$.

First step: adding a fresh instance

Let k_0 be the skeleton $\mathbb{A}_{n,r,C}(\mathbb{S}_{\sigma_{\text{id}}}(k))$ where C is a variable-disjoint renaming of the trace in r up to the ith event.[3] There is a homomorphism $k \overset{\phi_0,\sigma_0}{\longmapsto} k_0$ with $\sigma_0 = \sigma_{\text{id}}$ and ϕ_0 being nodewise bijective. In other words, (ϕ_0, σ_0) is an inclusion map.

Furthermore, (ϕ, σ) factors through (ϕ_0, σ_0). Let $\hat{\phi}$ be defined to be $\phi \circ \phi_0^{-1}$ for those nodes in the image of ϕ_0. Nodes not in the image of ϕ_0 are the nodes in the additional strand introduced by the augmentation. For such nodes (s, i), $\hat{\phi}$ maps them to (s', i) in k'. $\hat{\sigma}$ is defined to be σ on all variables appearing in $\phi_0(k)$. For the variables appearing outside the image of ϕ_0, note that there is a renaming ρ that maps the first i events in the trace of r to C used in our augmentation, and there is a substitution σ_r, the instantiation map in k', which unifies the first i events in the trace of r with the first i events in the strand s' in k'. (Specifically, let σ_r be the full instantiation map for that strand, restricted to the variables that appear in the first i events in the trace of r.) Then $\hat{\sigma}$ on variables appearing outside the image of ϕ_0 is $\sigma_r \circ \rho^{-1}$.

Second step: placing the critical message

Let t_r be the message sent in the ith node of the newly added strand in k_0. Note that $\hat{\sigma}$ unifies $t_r \mathbin{@} pp$ with tt, because $\hat{\sigma}(t_r) = \tau$ and $\tau \mathbin{@} pp = \hat{\sigma}(tt) = \sigma(tt)$. Let σ_1 be a most general unifier of $t_r \mathbin{@} pp$ and tt more general than $\hat{\sigma}$, and let $\hat{\sigma}_1$ be such that $\hat{\sigma} = \hat{\sigma}_1 \circ \sigma_1$.

Consider $k_1 = \mathbb{S}_{\sigma_1}(k_0)$. Let ϕ_1 be the identity map. Then (ϕ_1, σ_1) is a proto-homomorphism from k_0 into k_1.

[3] We can see that k_0 is a skeleton because no origination of values already in k changes, and because C is consistent with the origination restrictions introduced.

The second step in the augmentation is relies on parameters (k, n, r, i, pp, tt) where pp, the *placement position*, is a well-defined position in $C(i)$, and where tt, the *target term*, is either $t \mathbin{@} p$ where t is the message to be received at node n, or a member of $\{tt | tt$ is a subterm of an element of $E(\mathbf{P}_k(n), t \mathbin{@} p)\} \setminus E(\mathbf{P}_k(n), t \mathbin{@} p)$.[4] Here, we seek to find a preskeleton $aug_1(k, n, r, i, pp, tt)$ that has a new instance of role r at height i, that is as fresh as possible, subject to the constraint that $t' \mathbin{@} pp = tt$, where t' is the message sent in the new instance at height i.

We accomplish this, essentially, by considering $aug_0(k, n, r, i)$ where t' is the message sent in the new instance at height i, and finding $\sigma_1 \in unify(t' \mathbin{@} pp, tt)$ and then applying σ_1 to $aug_0(k, n, r, i)$ to produce $aug_1(k, n, r, i, pp, tt)$. This step can fail, if there is no unifier, or if $aug_1(k, n, r, i, pp, tt)$ is not a preskeleton, or if the map $(\phi_1 \circ \phi_0, \sigma_1 \circ \sigma_0)$ is not a preskeleton homomorphism.[5]

However, if $k \overset{\phi,\sigma}{\longmapsto} k'$ and k' has a strand s_r for which $|s_r| \geq i$, $rl(s_r) = r$, $(s_r, i) \prec \phi(n)$, and $t' \mathbin{@} pp = \sigma(tt)$ where t' is the message sent at (s_r, i), then (ϕ, σ) always factors through $(\phi_1 \circ \phi_0, \sigma_1 \circ \sigma_0)$ and this step does not fail.

Third step: ensuring carried-only-within

(complete me!)

□

Acknowledgments

The presentation of penetrator derivable messages in Section 5 is based on ideas by Javier Thayer.

References

[1] Joseph A. Goguen and Jose Meseguer. Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. *Theoretical Computer Science*, 105(2):217–273, 1992.

[4] Note that the latter will always include $t \mathbin{@} p$ itself unless the escape set is empty. It *is* possible for the escape set to be empty, in which case, explicitly specifying that $tt = t \mathbin{@} p$ is necessary.

[5] If, for instance, the unification requires identifying a variable occurring but not carried in $t' \mathbin{@} pp$ with a term in $N(k)$, and that variable appears earlier in the instance in a carried position, aug_1 would be a non-preskeleton.

[2] Joshua D. Guttman and F. Javier Thayer. Authentication tests and the structure of bundles. *Theor. Comput. Sci.*, 283(2):333–380, 2002.

[3] Alan Robinson and Andrei Voronkov. *Handbook of Automated Reasoning*. The MIT Press, 2001.

[4] F. Javier Thayer, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7(1), 1999.