Strand Spaces with Channels and Rules

Joshua D. Guttman John D. Ramsdell
The MITRE Corporation
CPSA Version 4.2.3

February 15, 2020

1 Introduction

This paper describes the formalism on which CPSA4 is built. Section 2 describes an implementation-oriented specification of strand spaces. It is based on the presentation for CPSA2 in [3], but modified so as to reflect changes implemented in CPSA4. In particular, channels and facts have been added to bundles and skeletons, and rules have been added to protocols.

In previous versions of CPSA, a communication event is either the transmission or reception of a message. In this version of CPSA, a communication event may optionally occur on a channel. If during a run of a protocol, the adversary never transmits a message on the channel, the channel is considered *authenticated*. Dually, if during a run of a protocol, the adversary never receives a message on the channel, the channel is considered to be *confidential*. A channel that is both authenticated and confidential is *secure*.

The messages without an associated channel—which are thus certainly not assumed to be authenticated or confidential—will be called *plain messages*.

Section 3 presents the strand spaces rule language that is used for rewriting. Protocols have been redefined so that they include both roles and rules.

Notation. A finite sequence f is a function from an initial segment of the natural numbers. The length of f is $|f|$, and $f = \langle f(0), \ldots, f(n-1) \rangle$ for $n = |f|$. When f is a sequence, we will write $g \colon |f| \to S$ to mean that $dom(g) = \{i \colon 0 \leq i < |f|\}$ and $ran(g) = S$. Thus, in effect we are regarding

each natural number i as being the set of natural numbers smaller than it, like the ordinals of ZF set theory.

If S is a set, then S^* is the set of finite sequences over S, and S^+ is the non-empty finite sequences over S. The prefix of sequence f of length n is $f \dagger n$. For partial function g, $g(x)\downarrow$ asserts g is defined at x.

2 Implementation-Oriented Strand Spaces

This section describes the formalism on which CPSA is based. The parameters to this strand space theory are:

1. a set of messages Alg. The set of messages Alg is the carrier set (or domain) of a term algebra.
2. a set of basic values $\mathsf{BV} \subset \mathsf{Alg}$. Keys and nonces are examples of basic values.
3. a *carried by* relation $\sqsubseteq \subseteq \mathsf{Alg} \times \mathsf{Alg}$. Intuitively, a message t_0 is carried by t_1, written $t_0 \sqsubseteq t_1$, if it is possible to extract t_0 from t_1 for someone who knows the relevant decryption keys.
4. a set of application specific predicate symbols P.

Example Message Algebra. The signature of one possible order-sorted [1] message algebra is in Figure 1. The algebra is the simplification of the CPSA message algebra used by the examples in this paper. In an order-sorted algebra, each variable x has a unique sort S. The *declaration* of x is $x : S$.

The algebra of interest is the order-sorted quotient term algebra generated by a set of declarations X. The message algebra Alg_X is the carrier set for sort M. The set of basic values BV_X is the union of the carrier sets for sorts A, S, and D. We write $t : S$ to say that term t is in the carrier set of sort S.

A variable has no intrinsic sort associated with it. Variables occur in the context of a set of declarations, and a declaration for the variable specifies its sort. The set of variables that occur in term t is $\mathit{Vars}(t)$. A variable declared to be of sort M is called a *message variable*.

A message t_0 is *carried by* t_1, written $t_0 \sqsubseteq t_1$, if t_0 can be derived from t_1 given the right set of keys. That is: $\sqsubseteq$ is the smallest reflexive, transitive relation such that

$$t_0 \sqsubseteq (t_0, t_1), \quad t_1 \sqsubseteq (t_0, t_1), \quad \text{and} \quad t_0 \sqsubseteq \{|t_0|\}_{t_1}.$$

Sorts:	M, A, S, D		
Subsorts:	$\mathsf{A} < \mathsf{M}$, $\mathsf{S} < \mathsf{M}$, $\mathsf{D} < \mathsf{M}$		
Operations:	$(\cdot,\cdot)$	$: \mathsf{M} \times \mathsf{M} \to \mathsf{M}$	Pairing
	$\{\!\|\cdot\|\!\}_{(\cdot)}$	$: \mathsf{M} \times \mathsf{A} \to \mathsf{M}$	Asymmetric encryption
	$\{\!\|\cdot\|\!\}_{(\cdot)}$	$: \mathsf{M} \times \mathsf{S} \to \mathsf{M}$	Symmetric encryption
	$(\cdot)^{-1}$	$: \mathsf{A} \to \mathsf{A}$	Asymmetric key inverse
	$\tau_0, \tau_1, \ldots$	$: \mathsf{M}$	Tag constants
Equations:	$(x^{-1})^{-1} = x$ for $x : \mathsf{A}$		

Figure 1: Simple Crypto Algebra Signature

Extra Sorts:	C, CM		
Operations:	$[\cdot]$	$: \mathsf{M} \to \mathsf{CM}$	Plain messages
	$[\cdot,\cdot]$	$: \mathsf{C} \times \mathsf{M} \to \mathsf{CM}$	Channeled messages

Figure 2: Channel Signature additions to a Message Signature

Channel Messages Some messages are transmitted over channels. A channel is a variable of sort C. For $c : \mathsf{C}$ and $t : \mathsf{M}$, $[c, t]$ associates message t with channel c, and is called a *channeled message*. A message t transmitted without a channel is written $[t]$, and is called a *plain message*. Fig. 2 shows the additions to a message signature required to support channels. The sort associated with a channeled or plain message is CM. The carried by relation is extended to channel messages as follows:

$$t \sqsubseteq [c, t] \text{ and } t \sqsubseteq [t].$$

Strand Spaces. A run of a protocol is viewed as an exchange of channel 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 finite non-empty sequence of *events*. An *event* is either a *channel message transmission* or a *channel message reception*. An event transmitting $m \in \overline{\mathsf{Alg}}$ is written as $+m$; and an event receiving channel message m is written as $-m$. If $e = \pm m$ is an event, then $msg(e) = m$. The set of traces over $\overline{\mathsf{Alg}}$ is $(\pm\overline{\mathsf{Alg}})^+$. When the context is clear, the plain message $[t]$ may be abbreviated by t.

1. A message t *originates* in trace a at index i iff $a(i) = +t_1$, $t \sqsubseteq t_1$, and for all $j < i$, $t \not\sqsubseteq msg(a(j))$.

2. A message t is *acquired* in trace a at index i iff $a(i) = -t_1$, $t \sqsubseteq t_1$, and for all $j < i$, t does not occur in $msg(a(j))$.

3. A variable x *appears* in trace a at index i iff x occurs in $msg(a(i))$, and for all $j < i$, x does not occur in $msg(a(j))$.

4. For algebra homomorphism σ,

$$\sigma \circ \langle \pm m_0, \ldots, \pm m_{n-1} \rangle = \langle \pm\sigma(m_0), \ldots, \pm\sigma(m_{n-1}) \rangle,$$

   so that $(\sigma \circ c) \dagger h = \sigma \circ (c \dagger h) = \sigma \circ c \dagger h$.

5. For traces a_1, a_2, $a_1 \star a_2 = \sigma$ if $\sigma(a_2 \dagger |a_1|) = a_1$, and the domain of σ is the variables that occur in $a_2 \dagger |a_1|$. When $a_1 \star a_2$ is defined, a_1 is said to be an *instance* of a_2.

6. A *strand space* is a set Σ of values, which we will call *the strands of* Σ, equipped with a *trace-of* operator $tr \colon \Sigma \to (\pm\overline{\mathsf{Alg}})^+$.

To avoid excess notation, we generally suppress tr and regard Σ as the strand space. We write $dom(\Sigma)$ for the underlying set of strands, and $\Sigma(s)$ for $tr(s)$ when $s \in dom(\Sigma)$.

The set of strands $dom(\Sigma)$ that CPSA uses are of a specific kind. They are finite initial segments of the natural numbers, so that Σ is in fact a sequence of traces.

Message events occur at *nodes* in a strand space. For each strand s, there is a node for every event in $\Sigma(s)$.

7. The *nodes* of strand space Σ are

$$\mathcal{N}(\Sigma) = \{(s, i) \mid s \in dom(\Sigma), i < |\Sigma(s)|\}$$

   and the event at a node is $evt_\Sigma(s, i) = \Sigma(s)(i)$.

8. The *message at* a node n, written $msg_\Sigma(n)$, is $msg(evt_\Sigma(n))$.

9. The *strand succession relation* $\Rightarrow$ is defined by

$$(s, i) \Rightarrow (s, i + 1) \text{ iff } s \in dom(\Sigma) \text{ and } i < |\Sigma(s)| - 1.$$

10. A basic value t is *non-originating in a strand space* Σ, written $non(\Sigma, t)$, iff it originates on no strand in Σ and each variable in t occurs in Σ.

11. A basic value t *uniquely originates at node n in strand space* Σ, written $uniq(\Sigma, t, n)$, iff t originates at index i in $s \in dom(\Sigma)$, $n = (s, i)$, and t originates on no other strand in $dom(\Sigma)$.

12. An atomic formula $p(t_1, \ldots, t_m)$ is a *fact of* Σ iff

    (a) $p \in P$ is a predicate symbol,

    (b) $t_i \in \overline{\mathsf{Alg}} \cup dom(\Sigma)$, and

    (c) each variable that occurs in t_i occurs in Σ.

Definition 1 (Bundle)**.** *Let $\mathcal{B} = (\Sigma, \rightarrow, \omega)$, where $\rightarrow \subseteq \mathcal{N}(\Sigma) \times \mathcal{N}(\Sigma)$ and ω is a set of atomic formulas. Relation $\rightarrow$ is called the* communication relation*. $\mathcal{B}$ is a* bundle *iff:*

1. *$n_0 \rightarrow n_1$ implies $evt_\Sigma(n_0) = +t$ and $evt_\Sigma(n_1) = -t$ for some message t;*

2. *$evt_\Sigma(n_1) = -t$ implies there exists a unique n_0 such that $n_0 \rightarrow n_1$;*

3. *$\mathcal{N}(\Sigma), \rightarrow \cup \Rightarrow$ forms a well-founded directed graph; and*

4. *for all $f \in \omega$, f is a fact of Σ.*

Let $\prec_{\mathcal{B}} = (\rightarrow \cup \Rightarrow)^+$, the transitive closure of the edges.

The transitive closure $\prec_{\mathcal{B}}$ is an irreflexive relation on $\mathcal{N}(\Sigma)$. This transitive, irreflexive relation specifies the causal ordering of nodes in a bundle. A transitive irreflexive binary relation is also called a strict (partial) order.

Since CPSA manipulates only finite strand spaces, and acyclicity is equivalent to well-foundedness for finite graphs, CPSA just checks acyclicity. Thus, in $\mathcal{B} = (\Sigma, \rightarrow, \omega)$, the nodes form the vertices of a directed acyclic graph, whose edges represent communications $\rightarrow$ and strand succession $\Rightarrow$ in Σ.

Lemma 1. *Let $\mathcal{B} = (\Sigma, \rightarrow, \omega)$ be a bundle. Then $\prec_{\mathcal{B}}$ is a well-founded strict partial order.*

If $S \subseteq \mathcal{N}(\Sigma)$ is non-empty, then S has $\prec_{\mathcal{B}}$-minimal members.

13. The strand space of bundle $\mathcal{B}$ is written $str(\mathcal{B})$, its communication relation is $\rightarrow_{\mathcal{B}}$, and the facts are $fac(\mathcal{B})$, so $\mathcal{B} = (str(\mathcal{B}), \rightarrow_{\mathcal{B}}, fac(\mathcal{B}))$.

Runs of Bare Protocols. In a run of a bare protocol, each strand is an instance of a role of that protocol. We view adversarial strands as constrained by roles, just like compliant, non-adversarial strands. Recall that when f is a sequence, $g\colon |f| \to S$ means that $dom(g) = \{i\colon 0 \leq i < |f|\}$.

Definition 2 (Role, Bare Protocol)**.**

1. *A* role *is of the form* $\mathsf{r}_X(a)$*, where*

   X *is the* parameters *of the role, the declarations used to generate its algebra.*

   $a \in (\pm\overline{\mathsf{Alg}})^+_X$ *is the trace of the role;*

   satisfying the following property:

   (a) *every message variable is acquired on* a*, i.e. its first occurrence is in a reception event of* a*;*

2. *The* listener role *is* $lsn = r_{\{x:\mathsf{M}\}}(\langle -x, +x\rangle)$.

3. *A* bare protocol *is a set of roles that includes the listener role.*

Notice that in one role, variable k may be declared to be of sort A, and in another, of sort S.

Definition 3. *A strand* $s \in \Sigma$ *is an* instance *of a role* $\mathsf{r}_Y(a)$ *in strand space* Σ *over* $\overline{\mathsf{Alg}}_X$ *iff the function* $inst(\Sigma, s, \mathsf{r}_Y(a))$ *is defined, and* $inst(\Sigma, s, \mathsf{r}_Y(a)) = \sigma$ *when*

1. $|\Sigma(s)| \leq |a|$*;*

2. $\sigma \in \overline{\mathsf{Alg}}_Y \to \overline{\mathsf{Alg}}_X$*; and*

3. $\sigma = \Sigma(s) \star a$.

We use the variables ρ, ρ', ρ_i, etc., to range over roles.

$$
\begin{array}{rll}
\mathit{create} = \langle +x \rangle & & x : \mathsf{A} \text{ or } x : \mathsf{S} \text{ or } x : \mathsf{D} \\
\mathit{pair} = \langle -x, -y, +(x, y) \rangle & & x, y : \mathsf{M} \\
\mathit{sep} = \langle -(x, y), +x, +y \rangle & & x, y : \mathsf{M} \\
\mathit{aenc} = \langle -x, -k, +\{\!|x|\!\}_k \rangle & & x : \mathsf{M} \text{ and } k : \mathsf{A} \\
\mathit{adec} = \langle -\{\!|x|\!\}_k, -k^{-1}, +x \rangle & & x : \mathsf{M} \text{ and } k : \mathsf{A} \\
\mathit{senc} = \langle -x, -k, +\{\!|x|\!\}_k \rangle & & x : \mathsf{M} \text{ and } k : \mathsf{S} \\
\mathit{sdec} = \langle -\{\!|x|\!\}_k, -k, +x \rangle & & x : \mathsf{M} \text{ and } k : \mathsf{S} \\
\mathit{unchan} = \langle -[c, t], +[t] \rangle & & c : \mathsf{C} \text{ and } t : \mathsf{M} \\
\mathit{chan} = \langle -[t], +[c, t] \rangle & & c : \mathsf{C} \text{ and } t : \mathsf{M}
\end{array}
$$

Figure 3: Adversary Roles

Adversary Model. The roles of adversarial behavior are in Figure 3. There are three *create* roles, one for each basic sort. In fact, the defining characteristic of a basic value is that it denotes the set of messages the adversary can create out of thin air, consistent with origination assumptions.

The adversary roles form the bare protocol Adv. We are always interested in bare protocols that contain these roles. Thus, given a set Π of roles that represent the legitimate, compliant behaviors of principals in some (real-world) bare protocol, we will refer to the roles of Π as the *regular* behaviors, and to strands that instantiate them as *regular strands*. We will be interested in the bundles that are executions of the "penetrated" bare protocol $\Pi^+ = \Pi \cup \mathsf{Adv}$.

14. A bundle $\mathcal{B} = (\Sigma, \rightarrow, \omega)$ is a *run of bare protocol* Π iff, for every $s \in \mathit{dom}(\Sigma)$, there is a role $\rho \in \Pi^+ = \Pi \cup \mathsf{Adv}$ such that $\mathit{inst}(\Sigma, s, \rho)\downarrow$.

15. $\mathit{Bnd}(\Pi)$ is the set of bundles that are runs of bare protocol Π.

16. For $\mathcal{B} \in \mathit{Bnd}(\Pi)$, $\varrho \in \mathit{dom}(\mathit{str}(\mathcal{B})) \rightarrow \Pi^+$ is a *role assignment* if for every $s \in \mathit{dom}(\mathit{str}(\mathcal{B}))$, $\mathit{inst}(\mathit{str}(\mathcal{B}), s, \varrho(s))\downarrow$.

Lemma 2. *If $\mathcal{B} \in \mathit{Bnd}(\Pi)$ and x is a message variable, then $x \notin \mathit{Vars}(\mathcal{B})$.*

Ensuring this is the purpose of requiring that in each role, every message variable in the trace is acquired.

Definition 4. *For bundle $\mathcal{B} = (\Sigma, \rightarrow, \omega)$, channel c is* authenticated *in $\mathcal{B}$, written $\mathit{auth}(\mathcal{B}, c)$, iff no strand in Σ is a full length instance of the chan*

adversary role instantiated with c. Dually, channel c is confidential *in $\mathcal{B}$, written conf$(\mathcal{B}, c)$, iff no strand in Σ is a full length instance of the unchan adversary role instantiated with c.*

Skeletons. A *skeleton* represents part of the regular behavior in a set of bundles. It consists of a strand space, a partial ordering on the nodes, assumptions about uncompromised keys and about freshly generated basic values, and a set of facts.

Definition 5. *$\mathbb{A} = \mathsf{k}_X(\Pi, \Sigma, \prec, \nu, \upsilon, \zeta, \chi, \omega)$ is a* skeleton of *bare protocol Π over declarations X iff:*

1. *A variable is declared in X iff it occurs in Σ.*
2. *For all $s < |\Sigma|$, there is some $\rho \in \Pi$ such that $inst(\Sigma, s, \rho)\downarrow$;*
3. *$\prec$ is a strict (partial) order on $\mathcal{N}(\Sigma)$;*
4. *for all $t \in \nu$, (i) $t \in \mathsf{BV}_X$; and (ii) for all $n \in \mathcal{N}(\Sigma)$, $t \not\sqsubseteq msg(n)$;*
5. *for all $(t, n) \in \upsilon$, (i) $t \in \mathsf{BV}_X$; (ii) for some $n = (s, i)$, t originates in $\Sigma(s)$ at i; and (iii) t originates at no other node in $\mathcal{N}(\Sigma)$.*
6. *for all $c \in \zeta$, c is channel that occurs in strand space Σ.*
7. *for all $c \in \chi$, c is channel that occurs in strand space Σ.*
8. *for all $f \in \omega$, f is a fact of Σ.*

We regard skeletons as giving us *partial information* about a set of bundles; these are the executions that it is compatible with. This notion of compatibility is determined by the the notion of a homomorphism, or information-preserving map between skeletons.

Definition 6 (Homomorphism)**.** *Let $\mathbb{A}_0 = \mathsf{k}_X(\Pi, \Sigma_0, \prec_0, \nu_0, \upsilon_0, \zeta_0, \chi_0, \omega_0)$ and $\mathbb{A}_1 = \mathsf{k}_Y(\Pi, \Sigma_1, \prec_1, \nu_1, \upsilon_1, \zeta_1, \chi_1, \omega_1)$ be skeletons of bare protocol Π.*

$H = (\phi, \sigma)$ is a skeleton homomorphism*, written $H\colon \mathbb{A}_0 \hookrightarrow \mathbb{A}_1$, if ϕ and σ are maps with the following properties:*

1. *$\phi\colon dom(\Sigma_0) \to dom(\Sigma_1)$ maps strands of $\mathbb{A}_0$ into those of $\mathbb{A}_1$. We require $|\Sigma_0(s)| \leq |\Sigma_1(\phi(s))|$, and we regard ϕ also as mapping nodes by the rule $\phi((s, i)) = (\phi(s), i)$;*

2. $\sigma\colon \overline{\mathsf{Alg}}_X \to \overline{\mathsf{Alg}}_Y$ *is a message algebra homomorphism;*

3. *for all* $n \in \mathcal{N}(\Sigma_0)$, $\sigma(evt_{\Sigma_0}(n)) = evt_{\Sigma_1}(\phi(n))$;

4. $n_0 \prec_0 n_1$ *implies* $\phi(n_0) \prec_1 \phi(n_1)$;

5. $\sigma(\nu_0) \subseteq \nu_1$;

6. $(t, n) \in \upsilon_0$ *implies* $(\sigma(t), \phi(n)) \in \upsilon_1$; *and*

7. $\sigma(\zeta_0) \subseteq \zeta_1$;

8. $\sigma(\chi_0) \subseteq \chi_1$;

9. $p(t_1, \ldots, t_m) \in \omega_0$ *implies* $p(\delta(t_1), \ldots, \delta(t_m)) \in \omega_1$, *where*

$$\delta(t) = \begin{cases} \sigma(t) & \text{if } t \in \mathsf{Alg} \\ \phi(t) & \text{if } t \in dom\,\Sigma. \end{cases}$$

Property 6 says the node at which a basic value is declared to be uniquely originating is preserved by homomorphisms.

Definition 7 (Skeleton of Bundle)**.**

1. *Suppose that* $\mathcal{B}$ *is a bundle of bare protocol* $\Pi^+ = \Pi \cup \mathsf{Adv}$, *and* $\mathbb{A} = \mathsf{k}_X(\Pi, \Sigma, \prec, \nu, \upsilon, \zeta, \chi, \omega)$ *is a* Π*-skeleton.* $\mathbb{A}$ *is a skeleton of* $\mathcal{B}$, *written* $\mathbb{A} \in skels(\mathcal{B})$, *iff the strands of* $\mathcal{B}$ *can be permuted to form bundle* $\overline{\mathcal{B}}$ *such that* Σ *is a prefix of* $str(\overline{\mathcal{B}})$, *and:*

   (a) *the strands not in* Σ *are adversary strands;*

   (b) $\prec = \prec_{\overline{\mathcal{B}}} \cap (\mathcal{N}(\Sigma) \times \mathcal{N}(\Sigma))$;

   (c) $\nu = \{t \mid non(\overline{\mathcal{B}}, t)$ *and each variable in* t *occurs in* $\Sigma\}$;

   (d) $\upsilon = \{(t, n) \mid uniq(\overline{\mathcal{B}}, t, n) \wedge n \in \mathcal{N}(\Sigma)\}$; *and*

   (e) $\zeta = \{c \mid auth(\overline{\mathcal{B}}, c)$ *and variable* c *occurs in* $\Sigma\}$;

   (f) $\chi = \{c \mid conf(\overline{\mathcal{B}}, c)$ *and variable* c *occurs in* $\Sigma\}$;

   (g) $\omega = \{f \in fac(\overline{\mathcal{B}}) \mid f$ *is a fact of* $\Sigma\}$.

2. $[\![\mathbb{A}]\!] = \{\mathcal{B}\colon\ \exists \mathbb{B}\, \exists H\, \mathbb{B} \in skels(\mathcal{B}) \wedge H\colon \mathbb{A} \rightarrowtail \mathbb{B}\}$.

3 Strand Spaces Rule Language

This section describes an order-sorted first-order language intimately tied to strand spaces with protocols. Given a bare protocol Π, the strand spaces rule language is written $\mathcal{L}(\Pi)$. The signature of $\mathcal{L}(\Pi)$ includes the sorts, sort orderings, and the function symbols of the protocol's message algebra signature, such as the one in Figure 1. It also includes the strand sort Z, and no other additions.

The predicates in the signature of $\mathcal{L}(\Pi)$ will now be described. For each role $\mathsf{r}_X(a) \in \Pi$, there are height and parameter predicates. There are $|a|$ unary height predicates $\Pi[\mathsf{r}_X(a), h] : \mathsf{Z}$ with $1 \leq h \leq |a|$. Relative to skeleton $\mathbb{A}$, $\Pi[\rho, h](z)$ asserts that strand z in $\mathbb{A}$ is an instance of ρ and has a height of at least h. For each parameter $x : S \in X$, there is a binary parameter predicate $\Pi[\mathsf{r}_X(a), x] : \mathsf{Z} \times S$. Relative to skeleton $\mathbb{A}$, $\Pi[\rho, x](z, t)$ asserts that strand z in $\mathbb{A}$ is an instance of ρ in which x is instantiated as t. If x appears in a and index i, the height of strand z must be greater than i.

For each base sort B, there are unary predicates $\mathsf{non} : B$ and $\mathsf{uniq} : B$. $\mathsf{non}(t)$ asserts t is non-originating in $\mathbb{A}$ and $\mathsf{uniq}(t)$ asserts t uniquely originates in $\mathbb{A}$.

There exists two predicates about channels, $\mathsf{auth} : C$ and $\mathsf{conf} : C$. $\mathsf{auth}(c)$ asserts c is authenticated in $\mathbb{A}$ and $\mathsf{conf}(c)$ asserts c is confidential in $\mathbb{A}$.

Let m be the length of the longest role in Π. There are m^2 precedence predicates $\mathsf{prec}[i, j] : \mathsf{Z} \times \mathsf{Z}$ for $0 \leq i, j < m$. $\mathsf{prec}[i, j](x, y)$ asserts that node (x, i) is before node (y, j) in $\mathbb{A}$. There are $3m$ origin predicates $\mathsf{uniq\text{-}at}[i] : B \times \mathsf{Z}$, with $0 \leq i < m$ and B as before. $\mathsf{uniq\text{-}at}[i](t, z)$ asserts that t uniquely originates in $\mathbb{A}$ at node (z, i). For each fact predicate symbol p, $\mathsf{fact}[p](t_1, \ldots, t_m)$ asserts that $p(t_1, \ldots, t_m)$ is a fact of $\mathbb{A}$. Finally, equality is binary.

To improve the readability of formulas to follow, we write $\mathsf{prec}(x, i, y, j)$ for $\mathsf{prec}[i, j](x, y)$, $\mathsf{uniq\text{-}at}(t, z, i)$ for $\mathsf{uniq\text{-}at}[i](t, z)$, and $\mathsf{fact}(p, t_1, \ldots, t_m)$ for $\mathsf{fact}[p](t_1, \ldots, t_m)$.

Semantics of Strand Space Rule Formulas. When formula Φ is satisfied in skeleton $\mathbb{A}$ with order-sorted variable assignment α, we write $\mathbb{A}, \alpha \models \Phi$. For $x : S \in X$, $\alpha(x)$ is in the carrier set of Alg_X for sort S. For $x : \mathsf{Z}$, $\alpha(x) \in dom(\Sigma)$. We write $\bar{\alpha}$ when α is extended to terms in the obvious way. When sentence Φ is satisfied in skeleton $\mathbb{A}$, we write $\mathbb{A} \models \Phi$. The semantics of atomic formulas is given in Figure 4.

Let $\mathbb{A} = \mathsf{k}_X(\Pi, \Sigma, \prec, \nu, \upsilon, \zeta, \chi, \omega)$.

- $\mathbb{A}, \alpha \models \Pi[\mathsf{r}_X(a), h](z)$ iff $|\Sigma(s)| \geq h$ and $\Sigma(s) \dagger h \star a\downarrow$, where $s = \alpha(z)$.
- $\mathbb{A}, \alpha \models P[\mathsf{r}_X(a), x](z, t)$ iff $|\Sigma(s)| \geq h$, $\Sigma(s) \dagger h \star a = \sigma$, and $\sigma(x) = \bar{\alpha}(t)$, where $s = \alpha(z)$, x appears in a at index i, and $h = i + 1$.
- $\mathbb{A}, \alpha \models \mathsf{prec}(x, i, y, j)$ iff $(\alpha(x), i) \prec (\alpha(y), j)$.
- $\mathbb{A}, \alpha \models \mathsf{non}(t)$ iff $\bar{\alpha}(t) \in \nu$.
- $\mathbb{A}, \alpha \models \mathsf{uniq}(t)$ iff $(\bar{\alpha}(t), n) \in \upsilon$ for some node n.
- $\mathbb{A}, \alpha \models \mathsf{uniq\text{-}at}(t, z, i)$ iff $(\bar{\alpha}(t), (\alpha(z), i)) \in \upsilon$.
- $\mathbb{A}, \alpha \models \mathsf{auth}(c)$ iff $\alpha(c) \in \zeta$.
- $\mathbb{A}, \alpha \models \mathsf{conf}(c)$ iff $\alpha(c) \in \chi$.
- $\mathbb{A}, \alpha \models \mathsf{fact}(p, t_1, \ldots, t_m)$ iff $p(\bar{\alpha}(t_1), \ldots, \bar{\alpha}(t_m)) \in \omega$.
- $\mathbb{A}, \alpha \models y = z$ iff $\bar{\alpha}(y) = \bar{\alpha}(z)$.

Figure 4: Strand Space Rule Formula Semantics

Rule Syntax. Given a bare protocol Π, a rule is an order-sorted first-order sentence in $\mathcal{L}(\Pi)$ with a restricted syntax. A rule is an implication in which the antecedent is a conjunction of atomic formulas, and the conclusion is a disjunction of possibly existentially quantified conjunctions of atomic formulas.

Definition 8 (Rule)**.** *A* rule *is a sentence of the form*

$$\forall \vec{x}\, \Phi \supset \bigvee_i \exists \vec{y}_i\, \Psi_i, \text{ where}$$

1. Φ *and* Ψ_i *are conjunctions of atomic formulas,*
2. *each variable that occurs free in* $\vee_i \exists \vec{y}_i\, \Psi_i$*, occurs in* Φ*,*
3. *each strand variable in* $\vec{y}_i$ *occurs in a height formula in* Ψ_i*, and*
4. *each non-strand variable in* $\vec{y}_i$ *occurs in a parameter formula in* Ψ_i*.*

Definition 9 (Protocol)**.** $\Upsilon = (\Pi, A)$ *is a* protocol *iff* Π *is a bare protocol and* A *is a set of rules in* $\mathcal{L}(\Pi)$*.*

Definition 10 (Run of Protocol)**.** *Bundle* $\mathcal{B}$ *is a* run of protocol $\Upsilon = (\Pi, A)$ *iff* $\mathcal{B}$ *is a run of bare protocol* Π *and for all* $\mathbb{A} \in \mathit{skels}(\mathcal{B})$ *and rules* $\Phi \in A$*,* $\mathbb{A} \models \Phi$*.*

The strand spaces rule language is a descendent of the security goal language [2]. They are very similar, but differ in several ways.

1. Pairing and encryption is purposely omitted from the security goal language.
2. The security goal language has variables that range over nodes rather than strands as is the case for the strand spaces rule language.
3. Node variables that occur in the antecedent of a security goal sentence occur in a height predicate in the antecedent.
4. Non-node variables that occur in the antecedent of a sentence occur in a parameter predicate in the antecedent.
5. Facts are not part of the security goal language.

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.

[2] Joshua D. Guttman. Establishing and preserving protocol security goals. *Journal of Computer Security*, 22(2):201–267, 2014.

[3] John D. Ramsdell, Joshua D. Guttman, Moses D. Liskov, and Paul D. Rowe. *The CPSA Specification: A Reduction System for Searching for Shapes in Cryptographic Protocols*. The MITRE Corporation, 2009. In `https://github.com/ramsdell/cpsa` source distribution, `doc` directory.