Neutral Nodes in PVS

John D. Ramsdell

April 29, 2014

1 Message and State Model

Messages and states are modeled by elements of an order-sorted algebra [1]. An order-sorted algebra is a generalization of a many-sorted algebra in which sorts may be partially ordered. The carrier sets associated with ordered sorts are related by the subset relation.

Figure 1 shows the signature of the algebra used in this paper. Sort M is the sort of TPM machine states, sort L is the sort of transition labels, and sort $\top$ is the sort of all messages. Messages of sort A (asymmetric keys), sort S (symmetric keys), sort D (data), and sort E (text) are called *atoms*. Messages are atoms, tag constants, or constructed using encryption $\{\!|\cdot|\!\}_{(\cdot)}$, hashing $\#(\cdot)$, and pairing $(\cdot,\cdot)$, where the comma operation is right associative and parentheses are omitted when the context permits. The canonical representative for each element in the algebra is the term that contains the fewest number of occurrences of the inverse operation $(\cdot)^{-1}$.

The value in the PCR associated with a TPM state is the message given by the *pcr* function,

$$\begin{aligned} \mathit{pcr} &: \mathsf{M} \to \top \\ \mathit{pcr}(\mathsf{bt}) &= \mathsf{g}_0 \\ \mathit{pcr}(\mathsf{ex}(t, m)) &= \#(t, \mathit{pcr}(m)) \end{aligned}$$

The function *pcr* is injective.

The TPM transition relation is $\tau \subseteq \mathsf{M} \times \mathsf{L} \times \mathsf{M}$, where $(m, \mathsf{bl}, \mathsf{bt}) \in \tau$ (boot), $(m, \mathsf{el}(t), \mathsf{ex}(t, m)) \in \tau$ (extend), or $(m, \mathsf{ol}(\mathit{pcr}(m)), m) \in \tau$ (observe). An infinite sequence of states π_s and labels π_ℓ is a *path* if $\forall i \in \mathbb{N}.\, (\pi_s(i), \pi_\ell(i), \pi_s(i+1)) \in \tau$ where $\pi_s(0) = \mathsf{bt}$. Observation transitions are used with the TPM quote and decrypt operations.

Sorts:	M, L, $\top$, A, S, D, E		
Subsorts:	$\mathsf{A} < \top$, $\mathsf{S} < \top$, $\mathsf{D} < \top$, $\mathsf{E} < \top$		
Operations:	bt	$: \mathsf{M}$	Boot state
	ex	$: \top \times \mathsf{M} \to \mathsf{M}$	Extend state
	bl	$: \mathsf{L}$	Boot label
	el	$: \top \to \mathsf{L}$	Extend label
	ol	$: \top \to \mathsf{L}$	Observe label
	$(\cdot, \cdot)$	$: \top \times \top \to \top$	Pairing
	$\{\!\lvert \cdot \rvert\!\}_{(\cdot)}$	$: \top \times \mathsf{A} \to \top$	Asymmetric encryption
	$\{\!\lvert \cdot \rvert\!\}_{(\cdot)}$	$: \top \times \mathsf{S} \to \top$	Symmetric encryption
	$\#$	$: \top \to \top$	Hashing
	$(\cdot)^{-1}$	$: \mathsf{A} \to \mathsf{A}$	Asymmetric key inverse
	$(\cdot)^{-1}$	$: \mathsf{S} \to \mathsf{S}$	Symmetric key inverse
	$\mathsf{a}_i, \mathsf{b}_i$	$: \mathsf{A}$	Asymmetric key constants
	s_i	$: \mathsf{S}$	Symmetric key constants
	d_i	$: \mathsf{D}$	Data constants
	e_i	$: \mathsf{E}$	Text constants
	g_i	$: \top$	Tag constants
Equations:	${\mathsf{a}_i}^{-1} = \mathsf{b}_i \quad {\mathsf{b}_i}^{-1} = \mathsf{a}_i$		$(i \in \mathbb{N})$
	$\forall k : \mathsf{A}.\, (k^{-1})^{-1} = k$		$\forall k : \mathsf{S}.\, k^{-1} = k$

Figure 1: Crypto Algebra with State Signature

Useful properties about paths through state space are given by the Init Extend and Prefix Extend Lemmas.

Lemma 1 (Init Extend)**.**

$$\begin{array}{l}\forall \pi_s, t : \top, m : \mathsf{M}, k \in \mathbb{N}. \\ \quad \pi_s(k) = \mathsf{ex}(t, m) \text{ implies} \\ \quad\quad \exists j \in \mathbb{N}.\, j < k \text{ and } \pi_s(j) = m \text{ and } \pi_s(j+1) = \mathsf{ex}(t, m)\end{array}$$

Lemma 2 (Prefix Extend)**.**

$$\begin{array}{l}\forall \pi_s, t : \top, i, k \in \mathbb{N}. \\ \quad i \leq k \text{ and } \pi_s(k) = \mathsf{ex}(t, m) \text{ implies} \\ \quad\quad \pi_s(i) \text{ is a subterm of } \pi_s(k) \text{ or} \\ \quad\quad \exists j \in \mathbb{N}.\, i \leq j < k \text{ and } \pi_s(j) = m \text{ and } \pi_s(j+1) = \mathsf{ex}(t, m)\end{array}$$

These two lemmas are used to prove the one employed to prove the security goal in the envelope protocol.

Lemma 3 (State Split)**.**

$$\begin{array}{l}\forall \pi_s, t, t_0, t_1 : \top, m : \mathsf{M}, i, k \in \mathbb{N}. \\ \quad i \leq k \text{ and } \pi_s(i) = \mathsf{ex}(t_0, \mathsf{ex}(t, m)) \text{ and} \\ \quad t_0 \neq t_1 \text{ and } \pi_s(k) = \mathsf{ex}(t_1, \mathsf{ex}(t, m)) \text{ implies} \\ \quad\quad \exists j_0, j_1 \in \mathbb{N}. \\ \quad\quad\quad j_0 < i < j_1 < k \text{ and } \pi_s(j_0) = \pi_s(j_1) = m \text{ and} \\ \quad\quad\quad \pi_s(j_0 + 1) = \pi_s(j_1 + 1) = \mathsf{ex}(t, m)\end{array}$$

2 Strand Spaces With State Synchronization

In strand space theory, the *trace* of a strand is a linearly ordered sequence of events $e_0 \Rightarrow \cdots \Rightarrow e_{n-1}$, and an *event* is a message transmission $+t$ or a reception $-t$, where t has sort $\top$, or a state synchronization $\circ\ell$, where ℓ has sort L. A *strand space* Θ is a map from a set of strands to a set of traces. We choose the set of strands to be a prefix of the natural numbers, so a strand space is finite sequence of traces.

A node names an event in a strand space. The set of *nodes* of strand space Θ is $\{(s, i) \mid s \in Dom(\Theta), 0 \leq i < |\Theta(s)|\}$, and the event at a node is $evt_\Theta(s, i) = \Theta(s)(i)$. A node is a *neutral node* in Θ if the event at the node

is a state synchronization. The set of nodes of Θ is $nd(\Theta)$ and the set of neutral nodes of Θ is $nd^\circ(\Theta)$.

A message t_0 is *carried by* t_1, written $t_0 \sqsubseteq t_1$ if t_0 can be extracted from a reception of t_1, assuming plaintext is extractable from encryptions. In other words, $\sqsubseteq$ is the smallest reflexive, transitive relation such that $t_0 \sqsubseteq t_0$, $t_0 \sqsubseteq (t_0, t_1)$, $t_1 \sqsubseteq (t_0, t_1)$, and $t_0 \sqsubseteq \{|t_0|\}_{t_1}$.

A message *originates* in trace c at index i if it is carried by $c(i)$, $c(i)$ is a transmission, and it is not carried by any messaging event earlier in the trace. A message t is *non-originating* in a strand space Θ, written $non(\Theta, t)$, if it originates on no strand. A message t *uniquely originates* in a strand space Θ at node n, written $uniq(\Theta, t, n)$, if it originates in the trace of exactly one strand s at index i, and $n = (s, i)$.

The model of execution is a bundle. The triple $\Upsilon = (\Theta, \rightarrow, \rightsquigarrow)$ is a *bundle* if it defines a finite directed acyclic graph, where the vertices are the nodes of Θ, and an edge represents communication ($\rightarrow$), state passing ($\rightsquigarrow$), or strand succession ($\Rightarrow$) in Θ. For communication, if $n_0 \rightarrow n_1$, then there is a message t such that $evt_\Theta(n_0) = +t$ and $evt_\Theta(n_1) = -t$. For each reception node n_1, there is a unique transmission node n_0 with $n_0 \rightarrow n_1$. For state passing, if $n_0 \rightsquigarrow n_1$, then n_0 and n_1 are neutral nodes, and transitions at neutral nodes are compatible with a path through state space. The transitions are *compatible* with a path if there is a one-to-one correspondence between neutral nodes and an initial segment of a path through state space, the state passing edges respect path ordering, and neutral node labels map to path labels. More formally, transitions at neutral nodes are state compatible [2, Def. 11] if there exists an $i \in \mathbb{N}$, $f \in nd^\circ(\Theta) \rightarrow \mathbb{Z}_i$, and path π such that

1. f is a bijection,
2. $\forall n_0, n_1 \in nd^\circ(\Theta).\, n_0 \rightsquigarrow n_1$ iff $f(n_0) + 1 = f(n_1)$, and
3. $\forall n \in nd^\circ(\Theta).\, evt_\Theta(n) = \pi_\ell(f(n))$

State compatibility implies that neutral nodes are totally ordered.

State compatibility Property 3 asserts that the label at a neutral node must agree with the label of some path through state space. It is this property that explains why a labeled transition system is used instead of a simple state transition system. The label asserts that a subset of the set of transitions are bound to the event at a neutral node. However, had a subset of the transition relation been bound to the event, all that could have been assert by Property 3 is that the path associated with the event is in the subset.

$$
\begin{aligned}
\mathit{create}(t : \mathsf{A}|\mathsf{S}|\mathsf{D}|\mathsf{E}) &= +t \qquad \mathit{tag}_i = +\mathsf{g}_i \\
\mathit{pair}(t_0 : \top, t_1 : \top) &= -t_0 \Rightarrow -t_1 \Rightarrow +(t_0, t_1) \\
\mathit{sep}(t_0 : \top, t_1 : \top) &= -(t_0, t_1) \Rightarrow +t_0 \Rightarrow +t_1 \\
\mathit{enc}(t : \top, k : \mathsf{A}|\mathsf{S}) &= -t \Rightarrow -k \Rightarrow +\{\!|t|\!\}_k \\
\mathit{dec}(t : \top, k : \mathsf{A}|\mathsf{S}) &= -\{\!|t|\!\}_k \Rightarrow -k^{-1} \Rightarrow +t
\end{aligned}
$$

Figure 2: Adversary Roles

Each acyclic graph has a transitive irreflexive relation $\prec$ on its vertices. The relation specifies the causal ordering of nodes in a bundle. A transitive irreflexive binary relation is also called a strict order. In a bundle, when $\prec$ is restricted to neutral nodes, it is can be shown that it is identical to $\rightsquigarrow^+$, where $\rightsquigarrow^+$ is the transitive closure of $\rightsquigarrow$. Furthermore, compatibility Property 2 can be used to derive the following relation between node ordering and the function f.

$$\forall n_0, n_1 \in nd^\circ(\Theta).\, n_0 \prec n_1 \text{ iff } f(n_0) < f(n_1)$$

In the remainder of this section, the theory of strand spaces used in the proofs has been simplified. In the full theory, origination assumptions can be inherited from roles. See [3] for all the gory details.

When a bundle is a run of a protocol, the behavior of each strand is constrained by a role. Adversarial strands are constrained by roles as are non-adversarial strands. A *protocol* is a set of roles, and a *role* is a function from message variables to a trace. A trace c is an *instance* of role r if c is a prefix of r applied to some messages. For protocol P, bundle $\Upsilon = (\Theta, \rightarrow, \rightsquigarrow)$ is a *run of protocol* P if there exists a role assignment $ra \in \mathit{Dom}(\Theta) \rightarrow P$ such that for all $s \in \mathit{Dom}(\Theta)$, $\Theta(s)$ is an instance of $ra(s)$. In what follows, we fix the protocol P and only consider bundles that are runs of P.

The roles that constrain adversarial behavior are in Figure 2. For the encryption related roles, $k : \mathsf{A}|\mathsf{S}$ asserts that k is either a symmetric or asymmetric key. For the create role, $t : \mathsf{A}|\mathsf{S}|\mathsf{D}|\mathsf{E}$ asserts that t is an atom.

3 Skeletons

In this paper, a skeleton will be specified using a sentence in order-sorted logic. The sorts are the message algebra sorts augmented with a sort Z for strands and sort N for nodes. The atomic formula $\mathsf{htin}(z, h, c)$ asserts

$$
\begin{array}{ll}
\Upsilon, \alpha \models x = y & \text{iff } \alpha(x) = \alpha(y); \\
(\Theta, \rightarrow, \rightsquigarrow), \alpha \models \mathsf{htin}(z, h, c) & \text{iff } |\Theta(\alpha(z))| \geq \alpha(h) \text{ and} \\
 & \quad \Theta(\alpha(z)) \text{ is a prefix of } \alpha(c); \\
\Upsilon, \alpha \models n_0 \ll n_1 & \text{iff } \alpha(n_0) \prec_\Upsilon \alpha(n_1); \\
(\Theta, \rightarrow, \rightsquigarrow), \alpha \models \mathsf{non}(t) & \text{iff } non(\Theta, \alpha(t)); \\
(\Theta, \rightarrow, \rightsquigarrow), \alpha \models \mathsf{uniq}(t, n) & \text{iff } uniq(\Theta, \alpha(t), \alpha(n)); \\
(\Theta, \rightarrow, \rightsquigarrow), \alpha \models \mathsf{sends}(n, t) & \text{iff } evt_\Theta(\alpha(n)) = +\alpha(t).
\end{array}
$$

Figure 3: Satisfaction

that strand z has a length of at least h, and its trace is a prefix of trace c. The formula $n_0 \ll n_1$ asserts node n_0 precedes node n_1. The formula $\mathsf{non}(t)$ asserts that message t is non-originating, and $\mathsf{uniq}(t, n)$ asserts that message t uniquely originates at node n. Finally, the formula $\mathsf{sends}(n, t)$ asserts that the event at node n is a transmission of message t. The roles of the protocol serve as function symbols.

For bundle Υ, variable assignment α, and formula Φ, satisfaction $\Upsilon, \alpha \models \Phi$ is defined using the semantics for atomic formulas specified in Figure 3. A bundle Υ is described by a skeleton iff the skeleton's sentence Φ is modeled by Υ, written $\Upsilon \models \Phi$.

References

[1] Joseph A. Goguen and José 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. State and progress in strand spaces: Proving fair exchange. *Journal of Automated Reasoning*, 48(2):159–195, 2012.

[3] John D. Ramsdell. Proving security goals with shape analysis sentences. Technical Report MTR130488, The MITRE Corporation, September 2013. `http://arxiv.org/abs/1403.3563`.