A Formal Semantics for SPKI

Jon Howell* and David Kotz

Dartmouth College, Hanover NH 03755, USA
{jonh,dfk}@cs.dartmouth.edu
http://www.cs.dartmouth.edu/

Abstract. We extend the logic and semantics of authorization due to Abadi, Lampson, et al. to support restricted delegation. Our formal model provides a simple interpretation for the variety of constructs in the Simple Public Key Infrastructure (SPKI), and lends intuition about possible extensions. We discuss both extensions that our semantics supports and extensions that it cautions against.

1 Introduction

This paper provides a formal semantics for the Simple Public Key Infrastructure (SPKI), an Internet Experimental Protocol [1]. The current (2.0) version of SPKI is a merger of SPKI 1.0 and the Simple Distributed Security Infrastructure (SDSI) 1.0.

SPKI is an elegant practical system that addresses the problem of ensuring that a user is *authorized* to perform an action, not just the problem of identifying the user. This focus allows for much more flexible sharing of resources through delegation; in contrast, systems based on authentication with a conventional public-key infrastructure (PKI) plus authorization with conventional ACLs limit the available modes of resource sharing. SPKI does incorporate a notion of authentication as well: its linked local namespaces bind keys to names. This notion of authentication is more general than conventional hierarchical PKI naming, allowing it to escape the "trusted-root" problem.

Unfortunately, SPKI is not founded on a formal semantics that can provide intuition for what it does, what it promises, what it assumes, and how it may or may not be safely extended.

Abadi, Lampson, and others defined an authorization system called the Logic of Authentication [2, 3]. This system provides delegation without restrictions. A user can encode restrictions by delegating control over "self as *role*" to another user, and adding the principal "self as role" to the ACL of the resource to be shared. The system is based on a formal semantics that explains how delegations interact with various combination operators for principals. Our formalism for SPKI is based on the semantics of the Logic of Authentication, extended to support restricted delegation and SPKI names.

Our formal treatment of SPKI is attractive for two reasons:

* Supported by the USENIX Association.

First, it supplies intuition for what SPKI operations mean. The proliferation of concrete concepts in SPKI can be understood as applications of just three abstractions: *principal*, *statement*, and *name*.

Second, the formalism gives us guidance in extending SPKI. We give examples of dangerous extensions that the formalism advises against, and we give examples of extensions that the formalism supports and that we use in our concrete system implementation.

2 Related work

Abadi provides a semantics for SPKI names [4], but its definition shares a flaw with that used for roles in the original logic [2]. We discuss Abadi's name semantics in Section 4.3.

Halpern and van der Meyden supply an alternate semantics for SPKI names [5], but it only encompasses the containment relation among names, and does not treat names as principals. As a result, it cannot relate names to compound principals nor relate names to other principals that are only connected by a restricted delegation.

Aura supplies a semantics for SPKI restricted delegation [6], but it is unsatisfying in that it essentially says what the reduction procedure says: a delegation is in place if there is a chain of delegation certificates and principals. It does not lend intuition about what the delegations mean. In contrast, our semantics connects restricted delegation to the logic of belief, a formal model that describes what a principal means when it delegates authority.

3 The logic and semantics of restricted delegation

In the original logic, a proposition s might mean "it would be good to read this file now." The statement A **says** s represents a principal A asserting the truth of s. A statement $B \Rightarrow A$ (read "B speaks for A") captures delegation from A to B: If B **says** s, then A agrees; we conclude A **says** s. As is conventional in the modal logic community, the symbol $\supset$ is used for logical implication. Table 1 summarizes the notation we use for sets in the following sections.

We assume here that the reader is familiar with the basic operation of modal logic. Hughes and Cresswell provide the canonical, concise introduction to modal logic [7]. Fagin et al. provide a gentler introduction with motivating examples [8]. The extended version of this paper [9] includes a brief introduction to each of the above topics, plus an overview of SPKI.

Lampson et al. mention in passing the idea of a restricted speaks-for operator [3, p. 272]. In this section, we introduce our *speaks-for-regarding* operator, which formalizes the notion of the restricted speaks-for operator. (The extended version of this paper proves the soundness of our axiomatization and the theorems mentioned here.) The new operator is written $B \overset{T}{\Rightarrow} A$, and read "$B$ speaks for

Set	*Example members*	*Description*
Σ	s, t	The set of primitive propositions. They represent resources.
Σ^*	σ, τ $s \wedge t$	The set of well-formed formulas (statements) constructed from Σ, $\wedge$, $\neg$, $\mathcal{A}\,\mathbf{says}$, and $\mathcal{B} \Rightarrow \mathcal{A}$
2^{Σ^*}	S, T, V	The set of sets of statements
P	A, B	The set of primitive principals. They represent agents, including people, machines, programs, and communications channels.
P^*	$\mathcal{A}, \mathcal{B}$ $A \wedge B$	The set of compound principals constructed from P, $\wedge$, $\vert$, and $\cdot N$
$\mathcal{N}$	N	The set of local names

Table 1. The symbols used to represent sets in this article.

A regarding the set of statements in T." T is any subset of Σ^*. The desired meaning is that when $\sigma \in T$,

$$B \overset{T}{\Rightarrow} A \supset ((B\,\mathbf{says}\,\sigma) \supset (A\,\mathbf{says}\,\sigma))$$

The power of the speaks-for-regarding operator $\overset{T}{\Rightarrow}$ is that A can delegate a subset of its authority *without modifying any ACLs*. Contrast the situation with the use of roles in the Logic of Authentication, where to delegate authority over a restricted subset of her resources, a user had to define a role and install that role in the ACLs of each resource to be shared.

Restricted speaks-for is transitive:

$$\vdash (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{B}) \wedge (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}) \qquad \text{(Axiom E1)}$$

We expect the $\wedge$ operation on principals to be monotonic over $\overset{T}{\Rightarrow}$:

$$\vdash (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \wedge \mathcal{C}) \overset{T}{\Rightarrow} (\mathcal{A} \wedge \mathcal{C}) \qquad \text{(Axiom E2)}$$

Restricted control over two principals is the same as restricted control over their conjunct:

$$\vdash (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}) \wedge (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{B}) \equiv \mathcal{C} \overset{T}{\Rightarrow} (\mathcal{A} \wedge \mathcal{B}) \qquad \text{(Axiom E3)}$$

Let $\mathcal{U}$ be the universe of all well-formed formulas; that is, those formulas over which a model $\mathcal{M}$ defines $\mathcal{E}$.[1] Restricted speaks-for degenerates to the original speaks-for when the restriction set is the set of all statements:

$$\vdash (\mathcal{B} \overset{\mathcal{U}}{\Rightarrow} \mathcal{A}) \equiv (\mathcal{B} \Rightarrow \mathcal{A}) \qquad \text{(Axiom E4)}$$

[1] $\mathcal{E}$ is the extension function as in the formalism of Abadi et al. The function maps each logical formula to the set of worlds where the formula is true.

If Bob speaks for Alice regarding a set of statements T, he surely speaks for her regarding a subset $T' \subseteq T$:

$$\vdash (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{T'}{\Rightarrow} \mathcal{A}) \qquad \text{(Axiom E5)}$$

Using Axiom E5, a chain of delegations can be collapsed to a single delegation, connecting the head principal in the chain to the tail, whose restriction set is the intersection of the restriction sets of each of the original delegations.

$$\vdash (\mathcal{C} \overset{S}{\Rightarrow} \mathcal{B}) \wedge (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{C} \overset{S \cap T}{\Rightarrow} \mathcal{A}) \qquad \text{(Theorem E6)}$$

This is not to say that $\mathcal{C}$ may not speak for $\mathcal{A}$ regarding more statements than those in the intersection; we address this topic further in Section 5.9.

If we have two restricted delegations from Alice to Bob, we might expect Alice to speak for Bob with respect to the union of the restriction sets. Because of the semantics we choose for $\overset{T}{\Rightarrow}$, however, this intuition does not hold.

$$(B \overset{S}{\Rightarrow} A) \wedge (B \overset{T}{\Rightarrow} A) \not\supset B \overset{S \cup T}{\Rightarrow} A \qquad \text{(Result E7)}$$

In the extended version of this paper, we describe a relation weaker than $\overset{T}{\Rightarrow}$ for which the intuitive statement holds.

The quoting operator ($|$) constructs compund principals such as $B|A$, read "B quoting A." When principal $B|A$ **says** σ, we conclude that B **says** (A **says** σ): B is asserting what he thinks A believes. The quoting operator is monotonic in both arguments over $\Rightarrow$. Quoting is still monotonic over $\overset{T}{\Rightarrow}$ in its left argument:

$$\vdash (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset \mathcal{C}|\mathcal{B} \overset{T}{\Rightarrow} \mathcal{C}|\mathcal{A} \qquad \text{(Axiom E8)}$$

Our semantics does not justify monotonicity in the right argument, however:

$$(\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \not\supset \mathcal{B}|\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}|\mathcal{C} \qquad \text{(Result E9)}$$

Hence, when quoting others, principals cannot automatically invoke the same delegated authority they have when speaking directly. The same counterexample that shows Result E9 shows the same property for the weak speaks-for-regarding relation defined in the extended version of this paper, so it seems that the notion of quoting simply does not mix easily with restricted delegation. This result appears to limit the usefulness of quoting, because principals cannot employ quoting with the same ease as in the Logic of Authentication.

We can salvage some of the convenience of quoting, however, by propagating the quoted principal through the restriction set. Let T^* be the closure of T with respect to the propositional operators $\neg$ and $\wedge$: $T \subseteq T^*$, and if $\sigma, \tau \in T^*$, then $\neg\sigma \in T^*$ and $\sigma \wedge \tau \in T^*$. Furthermore let TC be the closure of T with respect to the modal operator $\mathcal{C}$ **says**: $T \subseteq TC$, and if $\sigma \in TC$, then ($\mathcal{C}$ **says** σ) $\in TC$. Now $(T^*)C$ is the modal closure applied to the propositional closure of some original set T. With these definitions, we can justify this axiom:

$$\vdash \left(\mathcal{B} \overset{(T^*)C}{\Rightarrow} \mathcal{A}\right) \supset \left(\mathcal{B}|\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}|\mathcal{C}\right) \qquad \text{(Axiom E10)}$$

When $T = \mathcal{U}$, this axiom reduces to showing right-monotonicity for the original speaks-for relation. This axiom means that $\mathcal{A}$'s restricted delegation to $\mathcal{B}$ must explicitly include any "quotes" of $\mathcal{C}$ about which it is willing to believe $\mathcal{B}$. It seems awkward, but it is a useful result. Why? Because in any possible-worlds semantics wherein $(\mathcal{B} \stackrel{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B}|\mathcal{C} \stackrel{T}{\Rightarrow} \mathcal{A}|\mathcal{C})$ for *all* principals $\mathcal{C}$, the relation representing $\mathcal{A}$ depends on every other principal relation. The introduction of malicious principals with cleverly-chosen relations into such a system can effectively expand T until $T = \mathcal{U}$.

3.1 Semantics of $\stackrel{T}{\Rightarrow}$

Like Abadi et al. [2], we use a semantics based on possible worlds, modeling a system with a *model* $\mathcal{M} = \langle W, w_0, I, J \rangle$. W is a set of possible worlds and $w_0 \in W$ the distinguished "real" world. The interpretation function I maps each primitive proposition to the worlds where it is true, and the interpretation function J maps each primitive principal to its possible-worlds visiblity relation.

The semantic definition of $\stackrel{T}{\Rightarrow}$ is based on the notion of *projecting* a model into a space where only the statements in set T are relevant. The idea behind this definition is that if one were to take the "quotient" of a model M with respect to the dual of T, the resulting model $\overline{M}$ would be concerned only with statements in T. $B \Rightarrow A$ in $\overline{M}$ should be equivalent to $B \stackrel{T}{\Rightarrow} A$ in the original model. The model $\overline{M}$ is a projection of M that only preserves information about statements in T.

We begin the construction by defining an equivalence relation $\cong_T$: $W \times W$ that relates two worlds whenever they agree on all statements in T:

$$w \cong_T w' \ \textbf{iff} \ (\forall \sigma \in T, w \in \mathcal{E}(\sigma) \ \textbf{iff} \ w' \in \mathcal{E}(\sigma)) \qquad \text{(Definition E11)}$$

Then we define the mapping $\phi_T : W \rightarrow \overline{W}$ that takes worlds from the original model to equivalence classes under $\cong_T$:

$$\phi_T(w) = \phi_T(w') \ \textbf{iff} \ w \cong_T w' \qquad \text{(Definition E12)}$$

The equivalence classes belong to a set $\overline{W} = 2^T$; notice that worlds (equivalence class representatives) in $\overline{M}$ cannot be confused with those in M. The extended version of this paper gives a construction of $\phi_T(w)$.

Next we extend ϕ_T to the function $\phi^w_T : 2^W \rightarrow 2^{\overline{W}}$ that maps a set of worlds $S_w \subseteq W$ to a set of equivalence class representatives in the projected model:

$$\phi^w_T(S_w) = \{\overline{w} \mid \exists w \in S_w, \ \overline{w} = \phi_T(w)\} \qquad \text{(Definition E13)}$$

We use bar notation ($\overline{w}$) to indicate an equivalence class representative (member of a world of a projected model) as opposed to a member of W in the original model.

We can now give a semantic definition of restricted delegation:

$$\mathcal{E}(\mathcal{B} \stackrel{T}{\Rightarrow} A) = \begin{cases} W \text{ if } \forall w_0 \left(\phi_T^w(\mathcal{R}(\mathcal{A})(w_0)) \subseteq \phi_T^w(\mathcal{R}(\mathcal{B})(w_0)) \right) \\ \varnothing \ \text{ otherwise} \end{cases} \qquad \text{(Definition E14)}$$

For the justifications of several of the axioms it is more convenient to shift the projection (ϕ) operation to one side of the subset relation. To do so, we define

$$\phi_T^+(R) = \{\langle w_0, w_1' \rangle \mid \exists w_1 \cong_T w_1',\ \langle w_0, w_1 \rangle \in R\} \qquad \text{(Definition E15)}$$

Think of ϕ_T^+ as a function that introduces as many edges as it can to a relation without disturbing its projection under T.

We can use ϕ_T^+ to give an equivalent definition of $\stackrel{T}{\Rightarrow}$:

$$\mathcal{E}(\mathcal{B} \stackrel{T}{\Rightarrow} A) = \begin{cases} W \text{ if } \mathcal{R}(\mathcal{A}) \subseteq \phi_T^+(\mathcal{R}(\mathcal{B})) \\ \varnothing \ \text{ otherwise} \end{cases} \qquad \text{(Definition E16)}$$

The symbolic gymnastics of moving the projection to the right side of the $\subseteq$ relation is equivalent to the definition in terms of ϕ_T^w, but it makes some of the proofs more concise. The extended version of this paper shows the equivalence.

A casual intuition for this definition is that ϕ_T projects from the full model M down to a model in which worlds are only distinguished if they differ with regard to the truth of statements in T. If we collapse away the accessibility arrows that do not say anything about what is happening in T, and A's relation is a subset of B's relation in the projection, then A believes everything B believes about statements in T. This intuition is exactly what we want for restricted delegation.

What happens if we take an alternative semantic definition for restricted delegation? We explore one seemingly-natural but undesirable alternative and two other interesting alternatives in the extended version of this paper.

3.2 Additional benefits of $\stackrel{T}{\Rightarrow}$

Introducing the $\stackrel{T}{\Rightarrow}$ operator to the logic not only provides the important feature of restricted delegation, but it simplifies the logic by replacing the *controls* operator, replacing roles, and providing a formal mechanism for the treatment of expiration times.

Supplanting *controls*. Now that we have the restricted speaks-for relation, we can dispense with the special *controls* operator for building ACLs.

Recall Abadi et al.'s special identity principal **1** [2, p. 718]. Because it believes only truth, $(\mathbf{1}\ \textbf{says}\ s) \supset s$ for all statements s. That is, there is an implicit principal that controls all statements. We can replace every statement of the form A *controls* s with an equivalent one: $A \stackrel{\{s\}}{\Rightarrow} \mathbf{1}$. This statement ensures that if $A\ \textbf{says}\ s$, then $\mathbf{1}\ \textbf{says}\ s$. Since the **1** relation only contains edges from a node to itself, a model can only satisfy this condition by selecting an actual world w_0 where s is true.

Supplanting roles. Roles as originally defined are attractive, but they have the significant difficulty that introducing a new restricted role R_2 involves finding all of the objects that role should be allowed to touch, and adding A **as** R_2 to each of those ACLs. When one of those objects does not allow ACL modifications by A, it is impossible for A to express the desired new role. The SPKI document gives a vivid example that shows how ACL management can become unwieldy [1, p. 17].

With the speaks-for-regarding relation, A can introduce a new role R_2 for itself by allowing $(A\,\mathbf{as}\,R_2) \overset{T_2}{\Rightarrow} A$. In fact, roles are no longer necessary at all, but the **as** and **for** operators, or operators like them, may still be useful for building tractable implementations.

Roles, as semantically defined by Abadi et al., can also have surprising consequences because they belong to a global "namespace." Imagine that both Alice and Bob use the role R_{user} in their ACLs. That means that the same relation $\mathcal{R}(R_{\text{user}})$ encodes both the way that $A\,\mathbf{as}\,R_{\text{user}}$ is weaker than A, and the way that $B\,\mathbf{as}\,R_{\text{user}}$ is weaker than B. In the extended version of this paper, we give a detailed example model that demonstrates this problem.

Formalizing statement expiration. Lampson et al. treat expiration times casually: "Each premise has a *lifetime*, and the lifetime of the conclusion, and therefore of the credentials, is the lifetime of the shortest-lived premise" [3, p. 270]. It is likely that a formal treatment of lifetimes would be time-consuming and unsurprising, but the lifetimes are an unsightly element glued onto an otherwise elegant logical framework. Fortunately, the $\overset{T}{\Rightarrow}$ relation allows us to dispense with lifetimes.

Recall from [3, p. 271, note 4] that primitive statements such as s are meant to encode some operation in a real system. Assume that each s describes not only an operation, but the effective time the operation is to take place.[2] Further, assume a restriction set T in a delegation $B \overset{T}{\Rightarrow} A$ includes restrictions on the times of the operations under consideration. After the last time allowed by the set, the delegation remains logically valid, but becomes useless in practice. Furthermore, restrictions on T can be more than expiration times; one can encode arbitrary temporal restrictions, such as only allowing a delegation to be valid on Friday afternoons.

4 The semantics of SPKI names

Recall from Section 3.2 how roles share a global "namespace," and the danger of crosstalk between applications of the same role. SPKI names have the same dangerous property: identical names have different meaning depending on the "scope" in which they appear. Hence treating names as roles will not do; we must extend the logic and semantics to model names.

[2] Like Lampson et al., we ignore the issue of securely providing loosely synchronized clocks.

We introduce to the logic a new set of primitive *names*, $\mathcal{N}$. We also extend principal expressions to include those of the form $\mathcal{A} \cdot N$, where $\mathcal{A}$ is an arbitrary principal expression and $N \in \mathcal{N}$. $\mathcal{A} \cdot N$ is read "$\mathcal{A}$'s N." For example, if Alice is represented by the logical principal A, and N_{barber} is the symbolic name "barber," then $A \cdot N_{\text{barber}}$ is a principal that represents "Alice's barber." That is, $A \cdot N_{\text{barber}}$ represents whoever it is that *Alice* defines as her barber. Should Bob delegate authority to the principal $A \cdot N_{\text{barber}}$, he is relying on a level of symbolic indirection defined by Alice. Should Alice change who has authority over $A \cdot N_{\text{barber}}$, she has redefined the subject of Bob's delegation.

Because $\cdot$ only accepts a principal as its left argument, there is no ambiguity in the order of operations; $\mathcal{A} \cdot N_1 \cdot N_2$ can only be parenthesized $(\mathcal{A} \cdot N_1) \cdot N_2$. For example, "Alice's barber's butcher" is "(Alice's barber)'s butcher." Parenthesizing the expression the other way, as "Alice's (barber's butcher)," is unnatural because it requires the ungrounded subexpression "(barber's butcher)."

4.1 The logic of names

What properties do we want names to have?

Local namespaces. First, a principal should control the meaning of any names defined relative to itself:

$$\begin{gathered}\forall \text{ principals } \mathcal{A},\ \text{names } N : \\ (\mathcal{A}\,\mathbf{says}\,(\mathcal{B} \stackrel{T}{\Rightarrow} \mathcal{A} \cdot N)) \supset (\mathcal{B} \stackrel{T}{\Rightarrow} \mathcal{A} \cdot N)\end{gathered}$$

We do not take this statement as an axiom for the same reason that Abadi, Lampson et al. do not accept the handoff axiom [2, p. 715], [3, p. 273]: our semantics does not support it in general. Instead, as with the handoff axiom, we allow the implementation to assume appropriate instances of it.

Left-monotonicity. Name application should be monotonic over speaks-for. If Alice binds her name "barber" to Bob, and Bob binds his name "butcher" to Charlie, then we want "Alice's barber's butcher" to be bound to Charlie.

$$\vdash (\mathcal{B} \Rightarrow \mathcal{A}) \supset (\mathcal{B} \cdot N \Rightarrow \mathcal{A} \cdot N) \qquad \text{(Axiom E17)}$$

Using this rule, we can write the following to capture the desired intuition:

$$\begin{aligned}&(\mathcal{B} \Rightarrow \mathcal{A} \cdot N_{\text{barber}}) \supset \\ &\qquad \mathcal{B} \cdot N_{\text{butcher}} \Rightarrow \mathcal{A} \cdot N_{\text{barber}} \cdot N_{\text{butcher}}\end{aligned}$$

Distributivity. We combine the following pair of results

$$\vdash (\mathcal{A} \wedge \mathcal{B}) \cdot N \Rightarrow (\mathcal{A} \cdot N) \wedge (\mathcal{B} \cdot N) \qquad \text{(Theorem E18)}$$
$$\vdash (\mathcal{A} \cdot N) \wedge (\mathcal{B} \cdot N) \Rightarrow (\mathcal{A} \wedge \mathcal{B}) \cdot N \qquad \text{(Axiom E19)}$$

to show that names distribute over principal conjunction:

$$\vdash (\mathcal{A} \wedge \mathcal{B}) \cdot N = (\mathcal{A} \cdot N) \wedge (\mathcal{B} \cdot N) \qquad \text{(Theorem E20)}$$

Here is a motivating example: If Alice has two doctors Emily and Fred, and Bob visits doctors Fred and George, then who is "(Alice and Bob)'s doctor?"

$$\begin{aligned} E &\Rightarrow A \cdot N_{\text{doctor}} \\ F &\Rightarrow A \cdot N_{\text{doctor}} \\ F &\Rightarrow B \cdot N_{\text{doctor}} \\ G &\Rightarrow B \cdot N_{\text{doctor}} \end{aligned}$$

Applying Theorem E20, we conclude:

$$F \Rightarrow (A \wedge B) \cdot N_{\text{doctor}}$$

That is, Fred is the only person who serves as both people's doctor.

No quoting axiom. The principal $(\mathcal{A}|\mathcal{B}) \cdot N$ can be written, but we have yet to find a meaningful intuitive interpretation for it. $(\mathcal{A}|\mathcal{B}) \cdot N$ bears no obvious relation to $(\mathcal{A} \cdot N)|(\mathcal{B} \cdot N)$, for example. We allow the principal in the logic, but we offer no axioms for extracting quoting from inside a name application.

Nonidempotence. Finally, application of names should not be always idempotent. Unless some other speaks-for statement causes it, there is no reason that "Bob's barber's barber" should speak for "Bob's barber." We were initially tempted to model name application $(\cdot)$ with role application, because roles satisfy Axiom E17; however, roles are idempotent. It may be the case that the application of a name can become idempotent; the extended version of this paper gives an example.

4.2 The semantics of names

Names and name application cannot be modeled with the roles and the quoting operator, because quoting a role is always idempotent. Furthermore, using the same role for multiple uses of the same name by different principals introduces crosstalk as described in Section 3.2.

Instead, we model names as follows. First, add a new element K to the tuple that defines a model. A model with naming consists of:

$$\mathcal{M} = \langle W, w_0, I, J, K \rangle$$

The new interpretation function $K : P \times \mathcal{N} \rightarrow 2^{W \times W}$ maps a primitive principal A and a name N to a relation. The idea is that principals only define the first level of names in their namespaces; all other names are consequences of chained first-level name definitions.

Next extend $\mathcal{R}$ to define the relations for principals formed through name application. We want to define $\mathcal{R}(\mathcal{A} \cdot N)$ as the intersection of several other sets, each requirement ensuring a desired property. The definition, however, would end up circular (at requirement (I), with equal principals) if it were expressed in terms of set intersection. Instead, we define $\mathcal{R}(\mathcal{A} \cdot N)$ as the largest relation (subset of $2^{W \times W}$) satisfying all of the following requirements:

$$\mathcal{R}(\mathcal{A} \cdot N) \subseteq \mathcal{R}(\mathcal{B} \cdot N) \qquad \text{(I)}$$
$$(\forall \mathcal{B} : \mathcal{R}(\mathcal{A}) \subseteq \mathcal{R}(\mathcal{B}))$$
$$\mathcal{R}(A \cdot N) \subseteq K(A, N) \qquad \text{(II)}$$
$$(\text{when } A \in P)$$
$$\mathcal{R}(\mathcal{A} \cdot N) \subseteq \mathcal{R}(\mathcal{B} \cdot N) \cup \mathcal{R}(\mathcal{C} \cdot N) \qquad \text{(III)}$$
$$(\text{when } \mathcal{A} = \mathcal{B} \wedge \mathcal{C})$$
$$(\text{Definition E21})$$

Requirement (I) supports Axiom E17. Requirement (II) applies only to primitive principals, and allows each primitive principal to introduce definitions for first-level names in that principal's namespace. A system implementing instances of the handoff rule does so conceptually by modifying $K(A, N)$. Requirement (III) only applies to principal expressions that are conjunctions, and justifies Theorem E20.

There is no question some such largest relation exists. Since each requirement is a subset relation, at least the empty set satisfies all three. There is an upper bound, since every relation is a subset of the finite set $W \times W$. Finally, the largest relation must be unique. If there were two such relations, then any element in one must belong to the other, since it belongs to every set on the right-hand side of a subset relation in the requirements, and we arrive at a contradiction.

In our semantics, as in Abadi's, left-monotonicity (Axiom E17) turns out to be surprisingly powerful. In the extended version of this paper, we consider how to temper it. Note also that Axiom E17 applies only to unrestricted delegation ($\Rightarrow$). In the extended paper we consider a stronger version of left-monotonicity, generalized to the restricted-speaks-for relation ($\overset{T}{\Rightarrow}$), and discuss why it is difficult to support semantically. Because Theorem E20 derives from Axiom E17, it is similarly limited to the unrestricted case.

4.3 Abadi's semantics for linked local namespaces

Abadi gives an alternate logic and semantics for SPKI-style linked local namespaces [4]. (He refers to SDSI, from which SPKI 2.0 derives.) Abadi's notation diverges from that used in the Logic of Authentication [2], but the semantics are the same. Table 2 helps translate the notation. Our semantics differs in three interesting ways.

Abadi's notation	*Our notation*
S	Σ
$\mu : S \times \mathcal{W} \to \{true, false\}$	$I : \Sigma \to 2^W$
$\rho : N \times \mathcal{W} \to 2^{\mathcal{W}}$	$K : P \times \mathcal{N} \to 2^{W \times W}$
$a \in \mathcal{W}$	$w \in W$
principals p, q	$\mathcal{A}, \mathcal{B} \in P^*$
$n \in N$	$N \in \mathcal{N}$
$[\![n]\!]_a = \rho(n, a)$	$\mathcal{R}(\mathcal{A} \cdot N)(w) = K(\mathcal{A}, N)(w)$
$[\![p' \mathtt{s}\ n]\!]_a$	$\mathcal{R}(\mathcal{A} \cdot N)(w)$

Table 2. A guide to translating between Abadi's notation and ours

First, SPKI has special global names, so that if N_G is a global name, $\mathcal{A} \cdot N_G = N_G$. The result is that the same syntactic construct can be used to bind a local name to another local name or to a globally-specified name. All names in linking statements are implicitly prefixed by the name of the speaking principal; but if the explicitly mentioned name is global, the prefix has no consequence. We consider this syntactic sugar, and leave it to an implementation to determine from explicit cues (such as a key specification or a SDSI global name with the special !! suffix) whether a mentioned principal should be interpreted as local to the speaker.

Second, Abadi's logic adopts the handoff rule for names, which he calls the "Linking" axiom. Here it is, translated to our terminology:

$$\mathcal{A}\,\mathbf{says}\,(\mathcal{B} \Rightarrow (\mathcal{A} \cdot N)) \supset (\mathcal{B} \Rightarrow (\mathcal{A} \cdot N))$$

He validates the axiom by the use of composition to model name application, with which we disagree.

Indeed, the third and most important way our semantics differs from Abadi's explains just why we disagree. Abadi's semantics models name application as quoting (composition). Each unqualified (local) name is mapped to a single relation. This property can introduce crosstalk between otherwise unconnected principals; recall the example from Section 3.2. Even when a name relation is not constrained to be a role, the same problem arises. For example, let N represent the name "doctor." Imagine that Bob assigns Charlie to be his doctor: $C \Rightarrow B|N$. This is fine; Charlie should be able to do some things on Bob's behalf, but not everything: If $B|N \overset{T}{\Rightarrow} B$, then Charlie can do the things in T.

Enter Alice, who is not only omniscient ($A = \mathbf{1}$), but serves as her own doctor ($A \Rightarrow A|N$). Abadi's semantics requires that $\mathcal{R}(\mathbf{1}) \circ \mathcal{R}(N) \subseteq \mathcal{R}(\mathbf{1})$. At worst, $\mathcal{R}(N) = \mathcal{R}(\mathbf{1})$, causing $B|N = B$, enabling Charlie's doctor to make investment decisions on Charlie's behalf. At best, $\mathcal{R}(N) \subset \mathcal{R}(\mathbf{1})$, and $B|N$ begins spouting off random statements, some of which may be in T, making Bob believe random statements. Our semantics escapes this fate by assigning to each use of a name its own relation, then ensuring the correct subset relationships remain among those relations.

In summary, defining a meaningful semantics to local applications of names from the same global namespace is nontrivial. Our semantics depends on an existential definition involving the "largest set satisfying the requirements," and is therefore more opaque than illuminating. Despite its limitations, we feel that it is better than an alternative that implies undesirable consequences.

5 Modeling SPKI

The original Logic of Authentication is useful because its principals are general enough to model several parts of a computing system, from users to trusted servers to communications channels. To formally model SPKI with our extended calculus, we first give a construction that models the delegation-control bit.

5.1 Delegation control

The SPKI document gives the motivation for including a delegation-control bit in SPKI certificates. We disagree with the argument and fall in favor of no delegation control, and for the same reasons as described in the document: delegation control is futile, and its use tempts users to divulge their keys or install signing oracles to subvert the restriction. Such subversion not only nullifies delegation control, but forfeits the benefits of auditability provided by requiring proofs of authorization. Despite our opinion, we present a construction that models delegation control.

To model the delegation-control feature we wish to split the **says** modality into two separate modalities: "utterance," which represents a principal actually making a statement, and is never automatically inherited by other principals, and "belief," which is inherited transitively just as **says** is. Not only is introducing a new logical modality clumsy, but it would require us to support a dubious axiom, undermining the simplicity of the semantics.

Instead, we resort to an equivalent construct: we split each "real" principal $\mathcal{A}$ we wish to model into subprincipals $\mathcal{A}_u$ and $\mathcal{A}_b$. $\mathcal{A}_u$ shall say only the things that $\mathcal{A}$ utters (statements that are actually signed by $\mathcal{A}$'s key; recall that all certificate-issuing principals in SPKI are keys), and $\mathcal{A}_b$ shall say all of the things that $\mathcal{A}$ believes. $\mathcal{A}$ may inherit her beliefs from other principals (because she has delegated to other subjects the authority to speak on her behalf), and furthermore $\mathcal{A}$ should believe anything she utters. This last condition replaces the clumsy axiom we wished to avoid; instead we enforce it by explicitly assuming the following statement for all principals $\mathcal{A}$ and statements s:

$$\vdash \mathcal{A}_u \,\mathbf{says}\, s \supset \mathcal{A}_b \,\mathbf{says}\, s \qquad \text{(Assumption E22)}$$

Certificates issued by a concrete principal A are statements uttered by A asserting things that A believes, so we model them as statements about A_b said by A_u. The desirable outcome is that no principal can delegate authority to make herself utter something (make A_u say something); she may only utter the statement directly (by signing it with her key).

5.2 Restriction

Recall that a SPKI 5-tuple includes five fields: issuer, subject, delegation-control bit, authorization, and validity dates. Let I and S represent the issuer and subject principals. Let T_A represent the set of primitive permissions represented by the authorization S-expression, and T_V the set of primitive permissions limited by the validity dates (assuming the effective-time encoding of Section 3.2). The 5-tuple can be represented this way if its delegation-control bit is set:

$$I_u \,\textbf{says}\, S_b \overset{T_A \cap T_V}{\Rightarrow} I_b$$

or this way if not:

$$I_u \,\textbf{says}\, S_u \overset{T_A \cap T_V}{\Rightarrow} I_b$$

A 4-tuple has a name field (N) and no authorization field or delegation-control bit. It would be encoded:

$$I_u \,\textbf{says}\, S_b \overset{T_V}{\Rightarrow} I_b \cdot N$$

It seems natural that a delegation bit is meaningless for a name binding, for in SPKI, a name principal can never utter a statement directly, only a key principal can. It is surprising, however, that SPKI name-binding certificates omit the authorization field. Why not allow a principal to say the following?

$$I_u \,\textbf{says}\, (S_b \overset{\{shampoo\}}{\Rightarrow} I_b \cdot N_{\text{barber}})$$

As it turns out, our semantics does not support such restricted name bindings (see Section 4.2).

5.3 Linked local namespaces

The subject principals in the keys above may be either keys (each directly represented by a primitive principal) or a string of names grounded in a key. Hence namespaces are "local" in that names are meaningless except relative to a globally unambiguous key; namespaces are "linked" in that the naming operation may be repeated: If $K_1 \cdot N_1$ resolves to K_2, then $K_1 \cdot N_1 \cdot N_2$ is the same as $K_2 \cdot N_2$, perhaps defined as some K_3.

We give a logic and semantics for linked local namespaces in Section 4. We model the SPKI name subject "george: (name fred sam)" with the principal expression $K_{\text{george}} \cdot N_{\text{``fred''}} \cdot N_{\text{``sam''}}$. Substituting the principal expression for S_b, a 4-tuple takes on the general appearance:

$$I_u \,\textbf{says}\, ((K_S \cdot N_1 \cdots N_k) \overset{T_V}{\Rightarrow} I_b \cdot N_0)$$

5.4 Threshold subjects

A threshold subject is a group of n principals who are authorized by a certificate only when k of the principals agree to the requested action. Such certificates are really just an abbreviation for a combinatorially-long $\binom{n}{k}$ list of conjunction statements. For example, a certificate with a 2-of-3 threshold subject naming principals P_1, P_2, and P_3 and an issuer A can be represented as:

$$P_1 \wedge P_2 \Rightarrow A$$
$$P_1 \wedge P_3 \Rightarrow A$$
$$P_2 \wedge P_3 \Rightarrow A$$

Hence the logic easily captures threshold subjects, although any tractable implementation would obviously want to work with them in their unexpanded form.

5.5 Auth tags

The "auth tags" used in authorization fields in SPKI represent sets of primitive statements. Therefore, we simply model them using mathematical sets.

5.6 Tuple reduction

The SPKI access-control decision procedure is called "tuple reduction." A request is granted if it can be shown that a collection of certificates reduce to authorize the request. The reduced tuple's subject must be the key that signed the request; the tuple's issuer must represent the server providing the requested service; and the specific request must belong to the authorization tag of the reduced tuple.

It is clear that tuple reduction is sound with respect to the extended logic. When 5- and 4-tuples are encoded in the logic as shown in Section 4 and Section 5.2, tuple-reduction simply constructs a proof from several applications of Theorem E6 and Axiom E17.

5.7 Validity conditions

An optional validity condition, such as a certificate revocation list, a timed revalidation list, or a one-time revalidation, can be encoded in the logic using a conjunction. For example, a certificate requiring a timed revalidation would be interpreted

$$A \textbf{ says } (B \wedge (R|H_1)) \Rightarrow A$$

to mean that the revalidation principal R must verify that this certificate (with hash H_1) is valid. Principal R signs a revalidation instrument I with a short validity interval T_V

$$R \textbf{ says } I \stackrel{T_V}{\Rightarrow} R$$

and a given revalidation instrument would agree with all valid outstanding certificates:

$$I \ \textbf{says}\ \mathbf{0} \Rightarrow I|H_1$$
$$I \ \textbf{says}\ \mathbf{0} \Rightarrow I|H_2$$
$$\vdots$$

The principal $\mathbf{0}$ has relation $\mathcal{R}(\mathbf{0}) = \varnothing$, so that every principal speaks for $\mathbf{0}$. Using the logic, we can reason that

$$\mathbf{0} \Rightarrow I|H_1 \stackrel{T_V}{\Rightarrow} R|H_1$$

and since $B = B \wedge \mathbf{0}$, $B \stackrel{T_V}{\Rightarrow} A$. Notice the treatment of a certificate's hash as a principal. In the logic, principals are general entities and can be used to represent many objects and actors.

Negative certificate revocation lists can be handled similarly; an implementation examining a revocation list would conclude $I \ \textbf{says}\ \mathbf{0} \Rightarrow I|H_1$ for any H_1 *not* present in the list.

One-time revalidations are meant to be interpreted as having a zero validity interval. A system verifying a request s creates a nonce E, understanding $E\,\textbf{says}\,s$, and sends it to the revalidator R. R replies with a statement meant to be interpreted

$$R \ \textbf{says}\ E \stackrel{\{s\}}{\Rightarrow} R|H_1$$

Now both B and $E \stackrel{\{s\}}{\Rightarrow} R|H_1$ say s, so $A\,\textbf{says}\,s$. Any future request of the same sort will require another revalidation, for its s will have a different effective time.

5.8 Safe extensions

Our semantics suggests that SPKI may be safely extended to support a variety of principals other than public keys. Channels protected by secret keys or a trusted computing base, for example, are easily modeled as principals in the logic. In the examples in this article, we represent principals with symbolic names. Real principals, however, are represented by some mechanism that can verify that a given request comes from a particular principal. Examples of mechanisms for authenticating users include the UID mechanism in Unix, the Kerberos authentication server, and public key cryptography. Lampson et al. show that many common system components can be modeled as principals [3].

Compound principals let us represent useful trust relationships other than delegation. A conjunct principal $(A \wedge B)$, for example, represents a principal that only believes σ when both A and B believe σ. Hence a delegation to a conjunct principal is analogous to a check that requires two signatures to cash. Conjunct principals are not first-class entities in SPKI, although they can appear as threshold subjects; an extended SPKI might exploit Theorem E20. Quoting principals are also missing from SPKI; Lampson et al. give nice examples showing how quoting can help a multiplexed server or communications channel differentiate when it is working on behalf of one client versus another [3, Sections 4.3, 6.1, 6.2, and 7.1]. Without quoting, such a server has permission to make statements for either client, so it must perform an access-control check in advance of relaying a client's statement. Quoting lets the multiplexed server defer the complete access-control decision to the final resource server that verifies the proof. The result is improved auditability, since the gateway's role in the transaction is recorded at the server, and a smaller trusted computing base, since only a tiny part of the gateway code must be correct to pass on the authorization decision to the server.

5.9 Dangerous extensions

In this section, we argue that SPKI auth tags should not be extended to represent logical negations. If $\mathcal{B}$ speaks for $\mathcal{A}$ regarding multiple restriction sets, the semantics suggest that $\mathcal{B}$ actually has some authority not explicitly mentioned in either set. For example,

$$(\mathcal{B} \overset{\{\sigma,\tau\}}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{\{\sigma\wedge\tau\}}{\Rightarrow} \mathcal{A}) \qquad \text{(Axiom E23)}$$

means that a principal believed on a set of statements is also believed on their conjuncts. This conclusion seems fairly natural, but it is interesting to note that a restriction set actually permits more statements than it represents explicitly.

With the semantics for restricted delegation we define in Section 3, not only does

$$(\mathcal{B} \overset{\{\sigma,\tau\}}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{\{\sigma\wedge\tau\}}{\Rightarrow} \mathcal{A}) \qquad \text{(Axiom E24)}$$

hold, but also:

$$(\mathcal{B} \overset{\{\sigma\}}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{\{\neg\sigma\}}{\Rightarrow} \mathcal{A}) \qquad \text{(Axiom E25)}$$

This result implies that given authority on a set of primitive statements, a principal also has authority on any propositional formula constructed from those statements. It is surprising, for even if only $\mathcal{B} \overset{\{s\}}{\Rightarrow} \mathcal{A}$ is explicitly granted, $\mathcal{B}$ can also cause $\mathcal{A}$ to say the negation of s.

Perhaps scarier still is that

$$\begin{aligned} \mathcal{B} \overset{\{\sigma\}}{\Rightarrow} \mathcal{A} &\supset \mathcal{B} \overset{\{\sigma,\neg\sigma\}}{\Rightarrow} \mathcal{A} \\ &\supset (\mathcal{B}\ \textbf{says}\ \text{false}) \supset (\mathcal{A}\ \textbf{says}\ \text{false}) \end{aligned}$$

The conclusion is the definition of Abadi's $\mapsto$ relation:

> "Intuitively, $\mathcal{A} \mapsto \mathcal{B}$ means that there is something that $\mathcal{A}$ can do (say *false*) that yields an arbitrarily strong statement by $\mathcal{B}$ (in fact, *false*). Thus, $\mathcal{A} \mapsto \mathcal{B}$ means that $\mathcal{A}$ is at least as powerful as $\mathcal{B}$ in practice." [2, p. 713]

With these semantics, one might fear that no restriction is actually meaningful. How might we escape it? We might abandon the **K** axiom ($\mathcal{A}$ **believes** $s \wedge \mathcal{A}$ **believes** $(s \supset t) \supset \mathcal{A}$ **believes** t), so that principals no longer believe every consequence of their beliefs. This option is undesirable because it cripples the logic to only operate outside the scope of belief operators.

A second option is to both disallow negative statements in restriction sets and to use the weaker $\mathcal{B} \xrightarrow{T} \mathcal{A}$ relation (described in the extended paper) instead of $\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}$ to model delegation.

A third option is to prevent principals from making contradictory statements. This is difficult in general in a distributed system. One approach is to prevent principals from making negative statements at all. SPKI takes this approach. Its tags, which represent both restriction sets and individual statements, cannot represent both a statement and its logical negation. We provide a formal treatment of tags in the extended version of this paper.

Another extension might be to allow SPKI name bindings (4-tuples) to include authorization restrictions. As mentioned in Section 4.2, our semantics suggests that this seemingly-natural extension has undesirable consequences.

We conclude that in certain dimensions, SPKI is as strong as it can be. Changing SPKI by allowing principals to make negative statements or by allowing negative statements in restriction sets would push SPKI "over the edge," making its restrictions meaningless. Those proposing to augment SPKI, or other systems based on a logic of restricted delegation such as that of Section 3, must be wary of this hazard.

6 Summary

We extend the Logic of Authentication and its underlying possible-worlds semantics to support restricted delegation, delegation control, and local namespaces. To define the semantics of restricted delegation, we project a model to a set of worlds distinguished only by statements in the restriction set. The resulting system provides intuition and a formal framework in which we reason about the current SPKI system and possible extensions to SPKI.

One of the advantages our formal framework is that it represents the many complicated features of SPKI with three simple concepts: *principal*, *statement*, and *name*. Features such as threshold subjects and on-line validations can be modeled with compound principals and idiomatic statements. The simplicity also suggests that SPKI may be safely integrated with systems with notions of "principal" other than SPKI's public keys; such principals are desirable because they can exploit fast local or secret-key-protected channels. The results are applied in just this way in a prototype system implementation [10].

Our formalism also warns of the danger of apparently-harmless extensions. In our semantics, allowing a principal to utter both a statement and its negation or allowing restricted delegation to a name binding would reduce restricted delegation to meaninglessness. It would be imprudent to so extend SPKI without developing an alternate semantics that gives the extension meaning. One might also assume that delegation over two sets of permissions should combine to represent a delegation over the union of the permissions, but Result E7 suggests that this is not the case.

Acknowledgements

Thanks to John Lamping, who patiently helped Jon understand logical proof systems and semantic models. Thanks also Jon Bredin, Valeria de Paiva, Mark Montague and Larry Gariepy for their discussions, which helped refine the idea. Thanks to the USENIX organization for funding our research on this topic.

References

1. Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylonen. SPKI certificate theory, October 1999. Internet RFC 2693.
2. Martín Abadi, Michael Burrows, Butler Lampson, and Gordon Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, September 1993.
3. Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: theory and practice. *ACM Transactions on Computer Systems*, 10(4):265–310, November 1992.
4. Martín Abadi. On SDSI's linked local name spaces. *Journal of Computer Security*, 6(1-2):3–21, 1998.
5. Joseph Y. Halpern and Ronald van der Meyden. A logic for SDSI's linked local name spaces. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, pages 111–122, 1999.
6. Tuomas Aura. On the structure of delegation networks. In *Proceedings of the Eleventh IEEE Computer Security Foundations Workshop*, pages 14–26, 1998.
7. G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.
8. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
9. Jon Howell and David Kotz. A Formal Semantics for SPKI. Technical Report TR2000-363, Dartmouth College, Computer Science, Hanover, NH, March 2000. Available at: `http://www.cs.dartmouth.edu/reports/abstracts/TR2000-363/`.
10. Jonathan R. Howell. *Naming and sharing resources across administrative boundaries*. PhD thesis, Department of Computer Science, Dartmouth College, 2000.