An Access-Control Calculus for Spanning Administrative Domains

Jon Howell[*]
David Kotz

Technical Report PCS-TR99-361
Department of Computer Science
Dartmouth College
Hanover, NH 03755-3510
{jonh,dfk}@cs.dartmouth.edu

Abstract

In our quest to give users uniform access to resources unimpeded by administrative boundaries, we discovered that we needed transitive sharing among users, with the possibility of restricted access along each sharing link. To achieve that goal, we extend Lampson *et al.*'s calculus for access control to support restricted delegations.

We discuss the advantages of our extension, including the simplification of constructs like ACLs and statement expiration. We also apply our extension to model the Simple Public Key Infrastructure and make suggestions about its future development. Our extended calculus exposes some surprising consequences in such systems that use restricted delegation.

1 Introduction

Users should be able to access resources uniformly, seamlessly crossing administrative boundaries when they are not relevant to the task the user is trying to accomplish. One of the most difficult aspects of achieving that goal is finding an architecture for security that spans administrative domains. Most security architectures are woven into a specific system model, most of which treat a "system" as a unit administrated locally, or at best, in a hierarchy of domains.

In the real world, however, sharing happens across administrative domains. In Figure 1, Alice has access to a resource, and wants to share some of her access with Bob, who further wishes to share part of his access with Charlie. Conventional systems, based on Access Control Lists (ACLs), make this sort of sharing very difficult. First, each remote user must appear as a principal in the domain where the resource lives. Second, to implement each restriction, each user in the chain would need to be able to modify the ACL of the original resource. In this paper, we present a system that can model such chains of delegation satisfactorily, regardless of how they traverse administrative domains, or of the fact that each user may restrict what he shares with the next.

1.1 Background

Distributed systems that support multiple administrative domains are typically organized as a hierarchy. It is a start, but in essence, the hierarchy itself is still one large administrative domain. Sharing with users outside the hierarchy has the same problems as systems that model only a single administrative domain. It is unlikely that we can expect all or most users and

[*] Supported by a research grant from the USENIX Association.

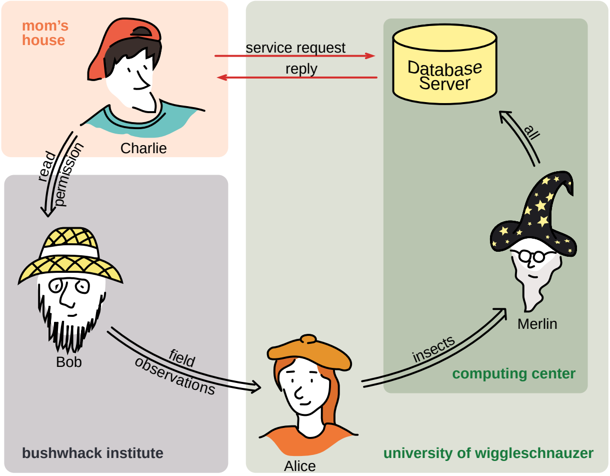


Figure 1: *A chain of delegation that spans administrative boundaries. The straight arrows represent actual communication, and the arcs represent delegations with restriction. Charlie's permission is limited by each of the restrictions on the path of delegation between himself and the database resource.*

organizations to subscribe to a single administrative hierarchy, especially when trust between distant parties requires putting trust in common root nodes: those nodes will have disproportionate power.

In the real world, users *do* share resources across administrative boundaries. Users need to share resources to do their work, so they find ways to move the resources. Imagine Bob has partial access to a resource owned by Alice. Bob may give Charlie a copy of a resource; but copies cannot support reference semantics. So Bob might share his identity, such as by sharing a password. Or he may set up an "oracle" that accesses the resource in the Bob's name, on Charlie's behalf, for example, by putting a local link to the resource into his publicly-visible ftp or http directory. All three methods prevent Alice from auditing the proxied access.

The calculus for access control described by Lampson, Abadi, Burrows, Wobber, and Plotkin is an excellent start toward a system that solves the transitive sharing problem [LABW92, ABLP93]. (For brevity, we refer to it as Lampson's calculus.) Their system is based in a formal logic, and can express several natural security concepts. Delegations in their system can span administrative domains, since they are between arbitrary principals. Auditing is preserved, because every access must ultimately be grounded in a chain of delegation showing the resource owner how the object came to be shared with the user making the present request. The calculus, however, has a critical limitation: restriction of access is based on ACLs. So even though users can share resources with others freely, they cannot restrict the sharing at each link. If the system cannot model the needed policy, users are likely to share without restriction for expediency, even when it is the wrong security decision.

The Simple Public Key Infrastructure (SPKI, pronounced "spooky") is a recent Internet Experimental Protocol [EFL$^+$99]. It too facilitates sharing across administrative boundaries, but it also dispenses with ACLs in favor of allowing restrictions along each sharing link. Unfortunately, SPKI is not presently based in a formal logic as is Lampson's calculus; it offers only a suggested implementation of a decision procedure. It also discards some of the expressivity of Lampson's calculus.

1.2 Contributions

We began this work by starting with Lampson's calculus, and extending it to include restricted delegation. The purpose of the extension was to inform a distributed system implementation that supports transitive sharing. Upon learning about SPKI, an implementation very close to the one we envisioned, we set out to use our extended calculus as an underpinning for SPKI, and to show how SPKI may be extended while retaining the confidence derived from a formal logic.

Our extension to Lampson's calculus has three important contributions. First, we support our goal of transitive sharing with restriction at each link. Second, the notion of ACLs go away, replaced by restricted delegation. Restriction that used to happen in an ACL is now expressed in the first link of a delegation chain, between the resource provider and the principal that would have been listed in the ACL with restricted permission. Third, delegation restrictions naturally model expirations and other time restrictions. Where Lampson's calculus has a separate mechanism to incorporate expirations, ours naturally incorporates expirations directly into the central concept of restriction.

By extending Lampson's calculus, we retain several useful features that we can then apply to current work, such as SPKI. First, we retain its formal nature, including a formal semantics that justifies the logic and any implementation based on the logic. The semantics provides an intuitive mathematical basis for the consequences of the system, as well as promise of consistency in the resultant logic. The semantics also suggests opportunities for consistent extensions, and warns us away from imprudent extensions. Second, the logic is simple. It has only four basic concepts: statements, principals, names, and restriction sets. A complex implementation such as SPKI may be mapped into these simple concepts, enhancing its comprehensibility. Third, we retain the notion of general principals, including those with only indirect representations. For example, quoting principals allow multiplexed resources to work for multiple users while deferring access-control decisions to a central location, minimizing the trusted computing base (TCB). This feature helps avoid traps such as Unix set-UID-root daemons that end up implementing access control decisions and extending the TCB.

1.3 Overview

This version of the paper begins with three sections of review material. We begin with a brief introduction to modal logic and its semantics in Section 2, followed by a review of Lampson's calculus and its notation in Section 3, and an overview of SPKI in Section 4. All three sections are presented in a tutorial style, but none of the work is our own. The erudite reader may skip any or all of these sections.

In Section 5 we introduce our restricted delegation extensions to Lampson's calculus, explore variations on it and paths not chosen, and discuss its advantages in more detail. In the following section, Section 6, we develop extensions to the logic and semantics to support SPKI-style linked local namespaces.

In Section 7, we cast SPKI into our extended calculus, showing some of the assumptions SPKI depends upon. Next, we explore the subtle consequences of restricted authorization in Section 8. Our own plans for using the calculus are outlined in Section 9. We discuss related work in Section 10, and summarize our results in Section 11. The Appendix provides proofs of claims appearing in the paper.

2 The logic of belief

> *The Sicilian smiled and stared at the wine goblets. "Now a great fool," he began, "would place the wine in his own goblet, because he would know that only another great fool would reach first for what he was given. I am clearly not a great fool, so I will clearly not reach for your wine."*
>
> *"That's your final choice?"*
>
> *"No. Because you knew I was not a great fool, so you would know that I would never fall for such a trick. You would count on it. So I will clearly not reach for mine either."* [Gol73, p. 157]

Modal logic is the logic of belief. One way to reason about permissions and sharing is to reason about who believes what. We call participants in a distributed system *agents*, and the symbols that represent agents in logical expressions *principals*. Principals can also represent sets of agents, or one agent quoting another; these are called *compound principals*, and we discuss them in Section 2.1. If Alice believes everything Bob believes (that is, Alice trusts Bob in every matter), then if Bob believes it is good to read a given file, Alice must believe the same. In this section, we develop a model for reasoning about logic in the presence of belief.

We begin with propositional logic. Assume there is a set of primitive (uninterpreted, independent) statements Σ.[1] For our purposes of

[1] Figure 4 provides a table of sets and variable notation used in this paper.

access control, we consider primitive statements such as "it is good to write to file X." This interpretation turns an imperative command into a declarative proposition. The primitive statements may be connected with *and* ($\wedge$) and *not* ($\neg$) to form arbitrary formulas. The *or* ($\vee$) and *implies* ($\supset$) operators are abbreviations for longer formulas made of $\wedge$ and $\neg$.

Next we introduce a *modal* operator **believes**.[2] If σ is a formula and principal A represents agent Alice, $A\,\mathbf{believes}\,\sigma$ is a formula that can be read "Alice believes σ is true." In time, we will introduce multiple **believes** operators, one per principal. For now, we would like to build a *model* that helps us understand which formulas A believes; that is, for which σ do we have $A\,\mathbf{believes}\,\sigma$?

To model this logic, we build a *Kripke structure.* A Kripke structure is a tuple of sets $\mathcal{M} = \langle W, I, J \rangle$. The members of set W represent *possible worlds.* The function I maps a primitive proposition (s) to the set of worlds where it is true, and the function J maps a principal to a relation on worlds in W. Together, I and J determine the truth value of every formula in every world in W; we describe them in more detail shortly.

First, some intuition: A principal A living in world w_0 considers some other set of worlds possible, and if a formula σ is true in each of those other worlds, then A believes the formula. The interesting thing about possible worlds is that the set of worlds A considers possible captures what she does not know: if a statement σ appears in one possible world and $\neg\sigma$ appears in another, then A knows neither σ nor $\neg\sigma$. As far as she is concerned, σ could go either way, because A cannot tell which of the possible worlds she actually is in.

When we write $\mathcal{M}, w_0 \models \sigma$ (pronounced "$\mathcal{M}$ at w_0 *models* σ"), we mean that in model $\mathcal{M}$ at world w_0, the formula σ is true. The mapping I tells us immediately about the truth of primitive propositions at different worlds, but we wish to determine the truth of arbitrary statements σ, including propositional connectives and our modal operators ($\sigma = A\,\mathbf{believes}\,\tau$). We illustrate with an example structure, shown in Figure 2.

The model contains three primitive statements, l, b, and p. The statement l means that our agent Alice (A) is in the produce department of a grocery store. Its negation, $\neg l$, means that Alice is in the meat department (it's a small store). The b primitive means that the store's bananas are yellow, and the p primitive means that the store's pork is fresh.

Recall the three parts of a model, $\langle W, I, J \rangle$. W is the set of possible worlds; in our case, since there are three primitive statements, there are at most eight: $W = \{w_0, w_1, \ldots w_7\}$. I is a relation that defines which primitive statements are true at which worlds. In our example, $I(b) = \{w_0, w_1, w_4, w_5\}$, since the bananas are only yellow in those four worlds. Finally, J is a function that maps principals to relations. Because we have only one principal (Alice), J has only one mapping, written $J(A)$. The relation $J(A)$ is depicted with arrows in the diagram. For example, $\langle w_0, w_1 \rangle \in J(A)$; that is, when the actual world is w_0, w_1 is a world Alice considers possible. In our example, it happens that Alice considers two worlds possible from each world.

Assume for a moment that the actual world is in fact w_0: Alice is in the produce department, the bananas are yellow and the pork is fresh. If Alice were omniscient, she would consider only w_0 possible, for that is indeed the state of things. Alice, however, is merely a shopper. She cannot see from the produce department what is going on in the meat department, and thus she cannot tell if the pork is fresh. She must also consider possible world w_1, where the pork is spoiled. She knows for certain her own location, though, so she can ignore worlds $w_4 \cdots w_7$. Because she is in the produce department and can see the bananas, she can also ignore worlds w_2 and w_3 in which the bananas are green.

We have explained the two arrows emanating from world w_0. The other arrows in the diagram, comprising the relation $J(A)$, communicate the same sort of information about any other state of affairs. For example, if the actual world were

[2] In conventional modal logic, $A\,\mathbf{believes}\,\sigma$ is written $\Box_A \sigma$.

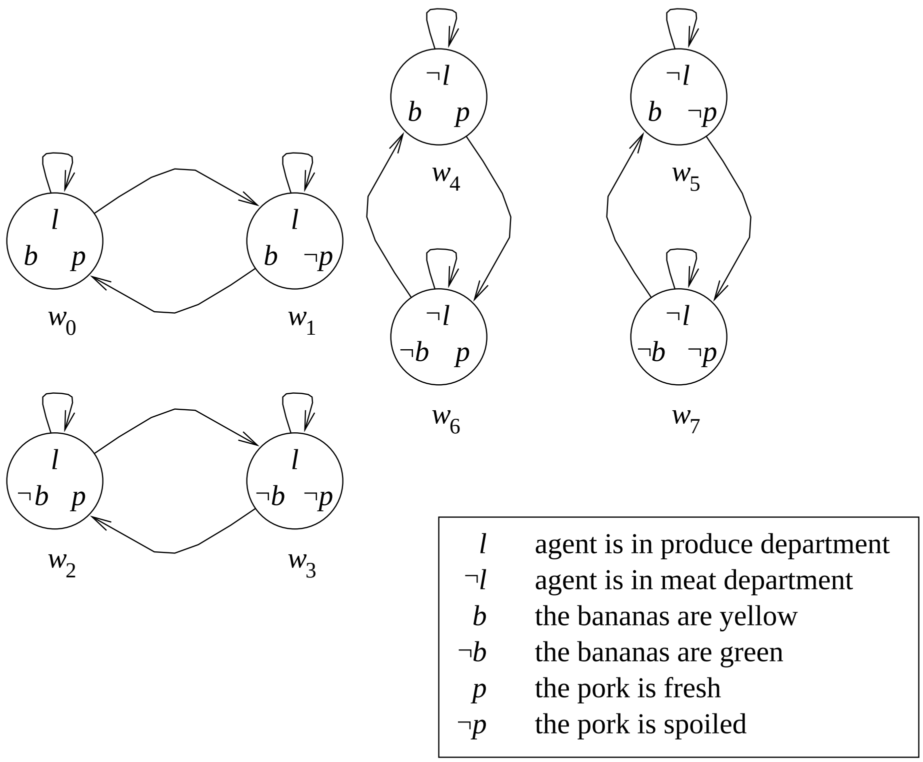


Figure 2: *A model of eight worlds (circles), illustrating the relationship between the accessibility relation (arrows) and the the modal operator (A **believes**).*

w_1 (the pork is in fact spoiled), Alice considers just the same worlds w_0 and w_1 possible, and for the same reasons.

Now that you have the intuition behind the Kripke structure, we can formally define when various statements are true. Primitive propositions are easy: the casual definition of I above becomes:

$$\mathcal{M}, w_0 \models s \qquad \text{when } w_0 \in I(s)$$

This definition can be read "Statement s is true at world w_0 in model M when w_0 is in the set $I(s)$."

What about formulas constructed from the propositional connectives $\wedge$ and $\neg$? The truth of some complex formula σ in a world is completely determined by the truth of its primitive propositions, which the model defines by the mapping I. So we can formally define an *extension* function $\mathcal{E}$ to extend the definition of I to arbitrary formulas. $\mathcal{E}$ is defined recursively starting with I, and extends as you would expect for the propositional connectives:

$$\begin{aligned}\mathcal{E}(s) &= I(s)\\ \mathcal{E}(\neg\sigma) &= W - \mathcal{E}(\sigma)\\ \mathcal{E}(\sigma \wedge \tau) &= \mathcal{E}(\sigma) \cap \mathcal{E}(\tau)\end{aligned}$$

Not surprisingly, $\neg\sigma$ holds in exactly those worlds where σ does not, and $\sigma \wedge \tau$ holds in exactly those worlds where both subformulas hold. Take a look at the example structure and convince yourself that $\mathcal{E}(b \wedge \neg p) = \{w_1, w_5\}$.

We embarked on this journey to discover when Alice believes various statements, so we need to find out when the model supports formulas including our modal belief operator. The natural intuition is that Alice should believe a statement whenever it is true in *every* world Alice considers possible. To recall our example, b is true (the bananas are yellow) in every world Alice considers possible from w_0, so $\mathcal{M}, w_0 \models A\,\textbf{believes}\,b$. But because Alice considers w_0 and w_1 possible, she considers both p and $\neg p$ possible; and so she can believe neither; hence we have $\neg(A\,\mathbf{believes}\,p)$ and $\neg(A\,\mathbf{believes}\,\neg p)$ at world w_0. (You can think of this situation as representing Alice's "silence" on the matter of p. Even though Alice asserts neither p nor $\neg p$, every formula is assigned a truth value. It is just that both $A\,\mathbf{believes}\,p$ and $A\,\mathbf{believes}\,\neg p$ are false.)

With this intuition, we fill out the definition of $\mathcal{E}$ to mention formulas containing our modal operator $A\,\mathbf{believes}$:

$$\mathcal{E}(A\,\mathbf{believes}\,\sigma) = \{w | J(A)(w) \subseteq \mathcal{E}(\sigma)\}$$

$J(A)(w)$ denotes the set of worlds that A considers possible from w.[3] So when σ is true in every one of these worlds (i.e., $J(A)(w) \subseteq \mathcal{E}(\sigma)$), then A believes σ (i.e. $A\,\mathbf{believes}\,\sigma$).

Of course, security is not very interesting in a world with only one agent. To introduce a second principal, we simply add a new relation $J(B)$ to our model. Now we can reason about what Bob believes ($B\,\mathbf{believes}\,\sigma$), and even about what Alice believes about what Bob believes ($A\,\mathbf{believes}\,B\,\mathbf{believes}\,\sigma$). (In our example, we could certainly discuss Alice's beliefs about her own beliefs, but for our application to access control, that is not very interesting.)

2.1 Compound principals

It is also possible to talk about *compound principals.* Lampson defines two operators on principals that can be used to make new compound principals. The first is fairly easy to describe: the principal $A \wedge B$ believes only things that both A and B believe. We can define a new possible-worlds relation for the compound principal in terms of the relations for A and B. To do this, we extend the mapping J to a new mapping $\mathcal{R}$ whose domain includes compound principals. Like the definition of $\mathcal{E}$, $\mathcal{R}$ is defined recursively starting with J:

$$\begin{aligned} \mathcal{R}(A) &= J(A) \\ &\qquad \forall \text{ primitive principals } A \\ \mathcal{R}(\mathcal{A} \wedge \mathcal{B}) &= \mathcal{R}(\mathcal{A}) \cup \mathcal{R}(\mathcal{B}) \\ &\qquad \forall \text{ arbitrary principals } \mathcal{A}, \mathcal{B} \end{aligned}$$

And R replaces J's role in the definition of $\mathcal{E}$:

$$\mathcal{E}(\mathcal{A}\,\mathbf{believes}\,\sigma) = \{w | \mathcal{R}(\mathcal{A})(w) \subseteq \mathcal{E}(\sigma)\}$$

That set union operation is surprising! What's going on? Recall that the more worlds an agent considers possible, the less the agent believes. In our example structure, Alice could not believe p because she considered world w_1 possible, where p was false. Likewise, by taking the union of the relations for principals A and B to get the relation for the compound principal $A \wedge B$, we ensure that the compound principal is at least as ignorant as either of A or B. If A and B disagree on any statement σ, then $A \wedge B$ can see both worlds where σ is true and worlds where it is false, so $A \wedge B$ can have neither belief.

The second operator for forming compound principals is written $B|A$, and pronounced "B *quoting* A." ("Quoting" may seem an odd choice of words when talking about belief; however, when we translate our terminology into that of Lampson *et al.*, it reads more naturally.) This principal captures B's beliefs about A's beliefs: $(B|A)\,\mathbf{believes}\,\sigma$ should be synonymous with $B\,\mathbf{believes}\,(A\,\mathbf{believes}\,\sigma)$.

The relation for the compound principal $B|A$ is the composition of the relations of B and A:

$$\mathcal{R}(B|A) = \mathcal{R}(B) \circ \mathcal{R}(A)$$

What is the intuition for using composition? Suppose we have $\mathcal{M}, w_0 \models B|A\,\mathbf{believes}\,\sigma$: At world w_0, Bob (agent B) believes Alice believes σ. That means that at every world Bob considers possible from w_0 ($\mathcal{R}(B)(w_0)$), Alice believes σ. But Alice only believes σ at those worlds if σ is true at every world Alice can see from those worlds:

$$\bigcup_{w' \in \mathcal{R}(B)(w_0)} \mathcal{R}(A)(w')$$

[3] Formally, $J(A)(w) = \{w' | \langle w, w' \rangle \in J(A)\}$.

The composition $\mathcal{R}(B) \circ \mathcal{R}(A)$ relates w_0 to just this set. So $B|A\, \mathbf{believes}\, \sigma$ is true at w_0 exactly when σ is true in every world reachable from w_0 by the composited relation given above as $\mathcal{R}(B|A)$.

2.1.1 The nature of principal relations

Now that we have a formal structure for discussing the beliefs of principals, let us consider what kinds of beliefs are reasonable, and how principals' beliefs should be related to one another's.

Recall our example structure, where in any world, Alice was either ignorant (had no belief) about either the pork or ignorant about the bananas. The first observation is that agents do not need to believe every true thing; statements about which they have neither a positive nor a negative belief represent something the agent is ignorant about.

Furthermore, observe that Alice never believed anything false: in every world, if $A\, \mathbf{believes}\, \sigma$, σ also held in that world. In the parlance of modal logic, we would say Alice's belief is actually *knowledge*: although she does not have all knowledge, everything she believes is in fact true. Why was this the case? Notice that Alice's possible-worlds relation is reflexive: for every world Alice's relation includes an edge pointing back to that world. That is why Alice cannot believe anything false. If σ is not true in a given world, Alice cannot believe σ there, because the definition

$$\begin{aligned} \mathcal{M}, w \models A\, \mathbf{believes}\, \sigma \;\; &\textbf{iff}\;\; w \in \mathcal{E}(A\, \mathbf{believes}\, \sigma) \\ &\textbf{iff}\;\; \mathcal{R}(A)(w) \subseteq \mathcal{E}(\sigma) \end{aligned}$$

precludes it.

In modeling access control in the presence of arbitrary principals, however, we should certainly expect that some principals will believe (or at least claim to believe) untrue things. So we make no restriction of reflexivity on the relation that defines a principal's beliefs. Indeed, a principal may have an empty relation at a world: it may consider *no* worlds possible! In that case, at that world, the agent considers every statement true, since every statement is true in all of the zero worlds the agent considers possible. Indeed, the agent believes *false*. The agent's reasoning has become inconsistent; other agents would be wise not to follow this agent's beliefs.

2.1.2 Trust

Agents following one another's beliefs is exactly how we model trust. If Alice establishes that she believes everything Bob believes, then Alice does not have to be present for Bob to read one of her files: if Bob claims that reading the file would be good, Alice must agree, and the file server grants the request. To capture this trust, we observe that Alice is "less ignorant" than Bob: she believes everything Bob believes, and then perhaps more (on which Bob may remain silent). Therefore, from any actual world, Alice should consider possible a subset of the worlds Bob considers possible. When $\mathcal{R}(A) \subseteq \mathcal{R}(B)$, Alice says everything Bob says; if she says even more, it is because she disregards some possible world that leaves Bob's belief ambiguous. You should convince yourself that if Bob believes σ, Alice has to believe the same thing, for she considers possible only a subset of the worlds Bob considers possible.

3 Lampson's calculus

This section contains an introduction to Lampson's calculus for access control. The reader familiar with it may skip to the next section. In the preceding section, we introduced an instance of modal logic: propositional logic plus some modal operators capture the possibly ignorant, possibly false beliefs of fallible principals. The semantics we presented, based on Kripke structures, is exactly that used by Abadi to justify the calculus for access control. We introduced the semantics first, though, because conventionally the semantics is the "intuitive model" of the world, and the logic is a system for discovering theorems (statements that are true in every model) and reasoning from premises to conclusions that must appear in the model.

To apply modal logic to access control, Lampson renames the operators. First, "believes" is renamed "**says**." This is meant to capture the notion that Lampson's logic is *performative*: sometimes when a principal says something, that something becomes true. The act of saying to a fileserver that a file should be modified, given that the fileserver believes you, causes that file to indeed be modified. This renaming makes the quoting operator sound more natural: $B|A$ is Bob quoting Alice. $B|A\,\textbf{says}\,s$ is meant to be a synonym for $B\,\textbf{says}\,A\,\textbf{says}\,s$. "Belief" is still useful intuition, however. The operator is the same; Bob's belief in σ can be inherited by Alice without Alice actually uttering σ.

A logic is a system of axioms and proof rules that let one reason from premises to conclusions: if the premise holds in a model, the conclusion holds as well. Lampson's logic is *sound* in that any conclusion proven in the logic holds in the model, but it is not *complete*: there are statements that are true in every model that cannot be proven in the logic. Abadi suggests that in fact the model may be *undecidable*: no logic system is adequate to prove every valid statement of the model.

The logic of access control is the same (up to variations in notation) as the conventional modal logic system K_n. The subscript n indicates that there are multiple modal operators [HC96, FHMV95, page 51]. We present that system here.

First, we write $\vdash \sigma$ if a statement σ is valid in the logic: either taken as an axiom, or provable as a theorem from other axioms and the proof rules. We prove theorems using the following:

If σ is a tautology of propositional calculus, then $\vdash \sigma$ (Axiom 1)

The axiom lets us pull in the theorems of propositional calculus without explicitly mentioning the axioms and proof rules that produce them.

$$\frac{\vdash \sigma \qquad \vdash \sigma \supset \tau}{\vdash \tau} \tag{Rule 2}$$

The proof rule (modus ponens) says that if both σ and the implication $\sigma \supset \tau$ are valid (provable), then τ is provable as well. It lets us prove theorems about formulas that include the modal operators (**says**) by reasoning from premises to conclusions.

We also have the *Distribution Axiom* (known in modal logic as the axiom **K**, from which the name of the system K_n derives):

$$\vdash A\,\textbf{says}\,(\sigma \supset \tau) \supset (A\,\textbf{says}\,\sigma \supset A\,\textbf{says}\,\tau) \tag{Axiom 3}$$

Intuitively it means that agents understand and believe all of the consequences of their beliefs. Furthermore, they believe every theorem:

$$\forall A,\ \frac{\vdash \sigma}{\vdash A\,\textbf{says}\,\sigma} \tag{Rule 4}$$

That is, agents know all of the theorems of the logic.

There is a subtle but important distinction between implication in the metalogic (the proof rule above) and implication in the logic. The logical symbol $\vdash$ means that the premises on its left prove the conclusions on its right. The proof rule condition $\vdash \sigma$ means that no premises are required to prove σ; that is, σ is a theorem. When that is true, we may conclude $\vdash A\,\textbf{says}\,\sigma$: it is proven that $A\,\textbf{says}\,\sigma$.

In contrast, the corresponding statement in the logic (not the metalogic) does not hold. The statement $\nvdash \sigma \supset A\,\textbf{says}\,\sigma$ is read "it is not provable that σ implies $A\,\textbf{says}\,\sigma$." The premise of the implication is an arbitrary statement σ (unlike the theorem $\vdash \sigma$ in the proof rule); it is not true that principals say every true statement. They say every theorem (those statements true in every world), but not every true statement (those statements true in the actual world from which the statement is being uttered).

3.1 The calculus of principals

The symbol $=$ is an equivalence relation on principals; by $A = B$ we mean that A and B have the same relation and therefore the same beliefs.[4]

[4] Abadi *et al.* "note that A and B can have the same beliefs without having the same possible worlds relation; however, because principals are identified by their rela-

(Later in the paper we also use $=$ to denote set equality; its use should be clear from context.)

We have presented the logical tools for reasoning about formulas of statements. Recall that we can also combine principals into principal formulas. For example, $A \wedge B$ is the principal that believes (says) only things that A and B agree upon. In the logic, $A \wedge B$ is defined in terms of its relationship to statements:

$$\vdash (A \wedge B)\,\mathbf{says}\,\sigma \equiv (A\,\mathbf{says}\,s) \wedge (B\,\mathbf{says}\,\sigma) \qquad \text{(Definition 5)}$$

Principal conjunction is associative, commutative, and idempotent:

$$\vdash (A \wedge B) \wedge C = A \wedge (B \wedge C) \qquad \text{(Axiom 6)}$$
$$\vdash A \wedge B = B \wedge A \qquad \text{(Axiom 7)}$$
$$\vdash A \wedge A = A \qquad \text{(Axiom 8)}$$

Quoting $(B|A)$ is defined as:

$$\vdash (B|A)\,\mathbf{says}\,\sigma \equiv B\,\mathbf{says}\ (A\,\mathbf{says}\,\sigma) \qquad \text{(Definition 9)}$$

In a sense, the quoting operator "curries" a **says** operation from the propositional formula into the principal formula, so that one can talk about a principal quoting another without yet mentioning the specific statement being quoted.

Quoting is associative and distributes over conjunction in both arguments:

$$\vdash (A|B)|C = A|(B|C) \qquad \text{(Axiom 10)}$$
$$\left.\begin{array}{l}\vdash A|(B \wedge C) = (A|B) \wedge (A|C) \\ \vdash (A \wedge B)|C = (A|C) \wedge (B|C)\end{array}\right. \qquad \text{(Axiom 11)}$$

3.2 The "speaks for" relation

A central concept of the calculus is the "speaks for" relation ($\Rightarrow$), which defines a partial order over all principals. This relation encodes the notion of one principal trusting another that we introduced in Section 2.1.2. The statement $B \Rightarrow A$ is read "B speaks for A," and means that whenever B says something, A certainly agrees. Formally, we define

$$\vdash (B \Rightarrow A) \equiv (B = B \wedge A) \qquad \text{(Definition 12)}$$

Why is this the case? If A trusts B, then A says everything B says. So the set of things $B \wedge A$ say must be the same as the set of things B says. It cannot be greater, by its semantic definition in Section 2.1, and it cannot be less, or else there is something B says that A does not.

From the definition we can derive:[5]

$$\vdash (B \Rightarrow A) \supset ((B\,\mathbf{says}\,\sigma) \supset (A\,\mathbf{says}\,\sigma)) \qquad \text{(Theorem 13)}$$

When $B \Rightarrow A$, B is a stronger principal than A in the sense that B can do everything A can do (by making A believe the appropriate performative statement), and perhaps more.

Using the associativity of $\wedge$ for principals, it is clear that $\Rightarrow$ is a transitive relation:

$$\vdash (B \Rightarrow A) \wedge (C \Rightarrow B) \supset C \Rightarrow A \qquad \text{(Theorem 14)}$$

(The $\wedge$ in the theorem is that for statements. We would like to use a different symbol for clarity, but we stick with Lampson's notation here.) Both the $\wedge$ and $|$ operators on principals are monotonic with respect to $\Rightarrow$:

$$\vdash (A \Rightarrow B) \supset ((A \wedge C) \Rightarrow (B \wedge C)) \qquad \text{(Axiom 15)}$$

$$\left.\begin{array}{l}\vdash (A \Rightarrow B) \supset ((A|C) \Rightarrow (B|C)) \\ \vdash (A \Rightarrow B) \supset ((C|A) \Rightarrow (C|B))\end{array}\right. \qquad \text{(Axiom 16)}$$

With the speaks-for relation, we can finally see why quoting is a useful operation. One can let $C|B \Rightarrow A$, so that C can only speak for A when it quotes B. Without quoting, we would need a formal accounting for universal quantification over formulas: $\forall\sigma, C\,\mathbf{says}\,B\,\mathbf{says}\,\sigma \supset A\,\mathbf{says}\,\sigma$.

tions in the semantics, we define equality in terms of relations." This is only possible if the model has two distinct worlds in W that belong to all the same I sets; that is, the model has two separate but indistinguishable worlds.

[5] Surprisingly, Abadi drops Definition 12 and instead treats Theorem 13 as an axiom. Doing so precludes theorems with conclusions containing $\Rightarrow$, since we are left with no axioms with $\Rightarrow$ in the conclusion. In fact, Theorem 13 requires only the weaker operator $\rightarrow$ in its premise, which we discuss in Section 5.1.2.

The semantics of $\Rightarrow$ falls out fairly directly. Definition 12 requires that

$$\begin{aligned} \mathcal{M}, w &\models B \Rightarrow A \\ &\textbf{iff } \mathcal{R}(B) = \mathcal{R}(B \wedge A) = \mathcal{R}(B) \cup \mathcal{R}(A) \\ &\textbf{iff } \mathcal{R}(A) \subseteq \mathcal{R}(B) \end{aligned}$$

Notice that the condition on the $\mathcal{R}$ relations is independent of the world w. So the extension function $\mathcal{E}$ is all-or-nothing for speaks-for formulas:

$$\mathcal{E}(B \Rightarrow A) = \begin{cases} W & \text{if } \mathcal{R}(A) \subseteq \mathcal{R}(B) \\ \varnothing & \text{otherwise} \end{cases}$$

(Definition 17)

3.3 Access Control Lists

The speaks-for relation, because it is transitive, lets us reason broadly about how principals' beliefs affect one another. In the end, however, the server wants to convince itself that some primitive proposition s, perhaps to be interpreted "it is okay to change the contents of the file," is true. To support this, Lampson uses the construct A *controls* s to indicate that principal A's beliefs about s are taken to be truth. It is defined as:

$$A \textit{ controls } s \equiv ((A \,\textsf{says}\, s) \supset s) \quad \text{(Definition 18)}$$

Now suppose B wants to write to the file that s describes, and the assumptions $\vdash B \Rightarrow A$ and $\vdash A$ *controls* s hold. Then the file server will be able to verify a proof of $\vdash s$, convincing itself that "it is okay to change the contents of the file."

Lampson encodes access control lists (ACLs) using *controls* assumptions:

$$\text{ACL } (O_1) = \left\{ \begin{array}{l} \vdash A \textit{ controls } s_{read}, \\ \vdash A \textit{ controls } s_{write}, \\ \vdash B \textit{ controls } s_{read} \end{array} \right\}$$

By adjusting which principals' assertions are believed, the ACLs allow or disallow agents to effect action.

3.4 Higher-level operators

The operating system that instantiates the calculus requires resource servers to construct and then verify all necessary proofs [WABL94]. Wobber calls it a *pull* model: it is the servers' job to pull in necessary assumptions and proof components needed to verify an agent's access. Building such proofs, when assumptions include speaks-for formulas with arbitrary combinations of $\wedge$ and $|$ operators, takes exponential time. To make the decision problem tractable, Lampson defines two high-level operators, **as** and **for**, in terms of the lower-level operators. Each operator is designed to reflect an idiomatic usage pattern of the calculus. By defining the higher-level operators, Lampson can restrict how the lower-level operators combine, and exploit characteristics such as associativity and idempotence. In the abstract, the operators can be treated as abbreviations and replaced by their definitions, and they do not affect the calculus. We cover them here to demonstrate the idioms they represent.

3.5 Roles and the " as " operator

Lampson defines a distinguished, disjoint set of principals called roles. By quoting a role, a principal restricts its own authority. For example, define the roles R_{user} and R_{admin} representing a person acting as a user and as an administrator, respectively. Suppose the ACLs in the system include $A|R_{\text{admin}}$ *controls* s_1 and $A|R_{\text{user}}$ *controls* s_2. In her daily work, Alice may step into her role as user by quoting R_{user}; when she needs to perform administrative tasks, Alice can explicitly quote R_{admin} to gain access to objects such as s_1 that mention her administrative role. More interestingly, Alice can delegate just one of her roles to another principal by arranging that $B \Rightarrow A|R_{\text{user}}$. Now Bob can do anything Alice could do as a user, but he cannot access her administrative resources. Roles can also be used to sandbox untrusted code. When running untrusted software, Alice might delegate to it only authority over $A|R_{\text{untrusted}}$, preventing the code from accessing the bulk of her resources.

The **as** operator stands for quoting when the quoted principal is a role. In a sense, **as** adds strong typing, requiring that its right-hand argument be a role. In contrast to general principals, quoting is idempotent and commutative for roles, and all principals automatically speak for themselves in every role:

$$R|R = R \qquad \forall R \in \mathit{Roles} \qquad \text{(Axiom 19)}$$
$$R'|R = R|R' \qquad \forall R, R' \in \mathit{Roles} \qquad \text{(Axiom 20)}$$
$$A \Rightarrow A\,\textbf{as}\,R \qquad \forall R \in \mathit{Roles} \qquad \text{(Axiom 21)}$$

By virtue of these special features of roles and its strong typing, the **as** operator takes on idempotence and commutativity. This helps make Lampson's access control problem tractable.

3.5.1 Semantics for Roles

The axioms above are not supported for general quoting, and yet **as** is simply an abbreviation for quoting. Therefore, the axioms must be justified by some restriction on the possible-worlds relations of the roles themselves. First we define a special principal **1**, the *identity*, who believes everything that is true and nothing that is not:

$$\mathcal{R}(\mathbf{1})(w) = w \qquad \forall w \in W$$

In any given world, **1** considers only that world possible. Therefore, it only tells the truth (the relation is reflexive), and it tells the whole truth (no world has multiple arrows, so it is confused about nothing). The identity serves as the most trusted role a principal can assume. Why? $A\,\textbf{as}\,\mathbf{1}$ is shorthand for $A|\mathbf{1}$, so $\mathcal{R}(A\ \textbf{as}\ \mathbf{1}) = \mathcal{R}(A) \circ \mathcal{R}(\mathbf{1}) = \mathcal{R}(A)$: the identity role does not limit A's authority at all.

All roles are principals whose relations are constrained as follows:

$$\mathcal{R}(R_1) \subseteq \mathcal{R}(\mathbf{1})$$

This means that the role relation may contain some edges $\langle w, w \rangle$ and not others, but no edges that take one world to another world. A role, when composed with another principal's relation, cannot expand the set of worlds the principal considers possible, only reduce it. See Figure 3 for an illustration.

We are now prepared to justify the axioms for roles. The first property is idempotence. $\mathcal{R}(R_1)$ takes each world to either itself or nowhere, so composing $\mathcal{R}(R_1)$ with itself should do the same. The second property is commutativity. An arrow appears in $\mathcal{R}(R_1) \circ \mathcal{R}(R_2)$ exactly when it appears in $\mathcal{R}(R_1) \cap \mathcal{R}(R_2)$, and $\cap$ is commutative. Finally, $A \Rightarrow A\,\textbf{as}\,R_1$ is automatically true when R_1 is a role. Why? Composing $\mathcal{R}(R_1)$ onto $\mathcal{R}(A)$ cannot introduce any new worlds (since the arrows of $\mathcal{R}(R_1)$ are all reflexive), but may eliminate worlds (when $\mathcal{R}(R_1)(w) = \varnothing$). Hence

$$\mathcal{R}(A) \circ \mathcal{R}(R_1) \subseteq \mathcal{R}(A)$$

and we conclude $A \Rightarrow A\,\textbf{as}\,R_1$.

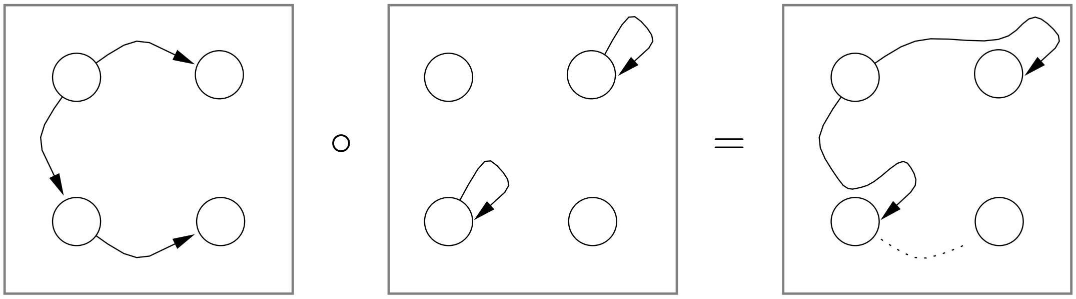
3.6 Delegation and the "for" operator

Besides encoding roles, quoting can be used to encode delegations to trusted principals in a restricted way. Here is the problem: Imagine that both Alice and Bob log in to machine M. Using just the speaks-for operator, Alice might establish that $M \Rightarrow A$ and Bob that $M \Rightarrow B$. But then when Bob (sitting at his terminal to machine M) tries to read a file that only A has permission to read, M would say the request, and the server would reason that A believed it. In this situation, the access-control system cannot help the server reason about whether the file should be read, since M has not provided enough information.

Instead, A could require that M explicitly mention A whenever it makes requests on A's behalf: $M|A \Rightarrow A$. Now when M is working for B, it will be quoting B, not A, and A's file is safe. If M were corrupt, of course, it could still abuse the authority granted it by A. But quoting principals helps an honest M pass the right information to resource servers for access-control decisions.

Lampson *et al.* define a slightly more complicated concept of delegation from A to B, written as the compound principal B **for** A. The key idea behind delegation is that both the delegator A and the delegate B must take some explicit action for the delegation to take effect:

$$A\ \textbf{says}\ B|A \Rightarrow (B\ \textbf{for}\ A)$$
$$B|A\ \textbf{says}\ B|A \Rightarrow (B\ \textbf{for}\ A)$$



An arbitrary principal relation $\mathcal{R}(A)$ ...

... composed with a role relation $\mathcal{R}(R)$ ...

... gives a new relation that is always a subset of $\mathcal{R}(A)$.

Figure 3: *Roles reduce relations they are composed with.*

from which, using the definition of **for** in Lampson's paper, we can conclude

$$(B|A) \Rightarrow (B\,\mathbf{for}\,A)$$

Then A installs $B\mathbf{for}A$ in ACLs for any resources it wishes to allow B to access on its behalf.

The difference between B simply taking care to always quote A and B receiving a delegation to B **for** A is subtle. In both cases, A must explicitly hand off authority to B. And in both cases, B has to take some explicit action to accept the delegation; in the first case, that action is to quote A, in the second, it must also make a separate statement accepting the delegation.

Like **as**, **for** seems to be introduced for its special properties, to enable a more efficient pull-style theorem-proving implementation.

We have completed our review presentation of the calculus due to Lampson *et al.*

4 The Simple Public Key Infrastructure

The Simple Public Key Infrastructure 2.0 (SPKI, pronounced "spooky") is an Internet Experimental Protocol created by Ellison, Frantz, Lampson, Rivest, Thomas, and Ylonen [EFL+99]. As its name suggests, it is designed to be a unifying standard for supporting public key authorization across the global Internet. We highlight here some of the features of SPKI relevant to this work.

First, SPKI's primary goal is to provide a server with evidence that the holder of a given cryptographic key is ultimately authorized for a request signed by that key. This goal contrasts with that of previous public-key infrastructure development efforts that attempted to bind keys to identities, and left authorization to be handled in the conventional fashion by ACLs that map identity to authorization.

In this section, we review the certificate semantics that SPKI supports, and outline the procedure used to determine whether a given certificate chain supports a requested operation.

4.1 Certificate semantics

To that end, SPKI defines its own certificate format, as well as an internal representation of certificates to which it can map other inputs, such as PGP certificates, X.509 certificates, or locally maintained ACL entries. Authorization results can be constructed from inputs providing information in one of three forms:

- $\langle \text{authorization}, \text{key} \rangle$
- $\langle \text{authorization}, \text{name} \rangle$
- $\langle \text{name}, \text{key} \rangle$

The first form coincides with SPKI's design philosophy of mapping keys directly to authorizations. Inputs of the latter two forms must ultimately be combined to form a SPKI $\langle \text{authorization}, \text{key} \rangle$ mapping.

Inputs of the first two forms are mapped into a data structure called a *5-tuple* for internal processing; inputs of the latter form are mapped into a data structure called a *4-tuple*.

A 5-tuple has the following fields:

- issuer: the public key granting the permission defined by the 5-tuple
- subject: a public key or name to which the permission is being granted
- delegation-control: a boolean value indicating whether this permission may be further delegated
- authorization: a set of primitive permissions being granted
- validity dates: a date range limiting the validity of this delegation

The intended meaning is that the issuer grants the subject the permission described in the authorization field for the duration of the validity dates. If the delegation-control bit is set, the subject may further delegate any or all of the permission to another subject.

The subjects in 5-tuples (and in 4-tuples, which we present shortly) may be replaced with a k-of-n threshold function. In this case, the permission is delegated to any principal that can prove it is authorized to speak for any k of the n "subordinate" subjects listed in the threshold function.

The authorization fields contain primitive permissions whose interpretation is left to the application employing the SPKI authorization engine. These permissions are represented using S-expressions. S-expressions encode infinitely large sets of primitive statements in a form that permits a compact representation of certain subsets. To a first approximation, primitives are trees, and an S-expression represents the subset of primitives that share a given "trunk" (our term). An S-expression contains the representation of a finite tree "trunk," and any primitive whose tree is formed of the trunk and some subtrees descending from the trunk's leaves is a member of the subset described by the S-expression. Notably, an S-expression can represent only a set of primitive symbols; never a formula made from the negation or conjunction of primitive symbols. S-expressions admit a simple intersection algorithm that always yields a compact representation of the intersected set: the "union" of two trunks is a new trunk that matches only primitive symbols that matched both input trunks.

SPKI certificates may also indicate an on-line mechanism for verifying that the issuer considers a certificate still valid. Two of the checks, the certificate revocation list (CRL, a negative list of revoked certificates) and the timed revalidation (a positive list of still-valid certificates), are performed by consulting a list revised more frequently than the original certificate being checked. The one-time revalidation check is performed by contacting a named server to verify that the server still approves the certificate, and "represents a validity interval of zero" [EFL$^+$99, p. 21].

Symbolic names are always interpreted relative to a globally unambiguous name; in SPKI, such a name always consists of a public key. As a consequence, the definition of a symbolic name is never ambiguous; it is always the definition supplied by the key that grounds the name. The SPKI authors contrast this situation with that of PGP, where symbolic names reside in a global namespace, and their meaning depends on the beholder and the "introducers" that the beholder trusts.

A symbolic name ultimately is defined as one or more keys, although a single 4-tuple may define a name in terms of a chain of other names grounded in a key. In that case, other 4-tuples must participate in the reduction of the name chain to a final key. A 4-tuple has the following fields:

- issuer: the public key defining this name in its private name space.
- name: the name being defined
- subject: a public key or name to which the name is bound.

- validity dates: a date range limiting the validity of this delegation

When the subject is a key, a 4-tuple binds a symbolic name to a key; when the subject is itself a name (always globally qualified), the 4-tuple binds a symbolic name to another name. In the latter case, the subject name must ultimately resolve to some key to be used in an authorization decision.

The intended meaning of a 4-tuple is that the issuer defines the symbolic name, when grounded by the issuer's key, to be equal to the key identified by the subject for the duration of the validity dates. It is easy to read this definition backwards. Note that a name definition tuple does not give the issuer control over the subject, but the subject control over any permission elsewhere granted to the grounded name "issuer: name." Hence a threshold subject is also meaningful as the subject of a 4-tuple; its use means that if a principal speaks for k of the n subordinate subjects, that principal also speaks for "issuer: name," and hence garners any permission granted to that name.

4.2 Tuple reduction

The SPKI access-control decision procedure is called "tuple reduction." Once the appropriate certificates for an access-control decision have been gathered and the on-line checks performed, and the certificates converted into internal tuples, the tuples are "reduced." If the reduction results in a 5-tuple granting the requested permission to the key that signed the request, then the request may be granted.

Reduction proceeds as follows. First, 4-tuples are reduced to resolve names. 4-tuples that define a name in terms of another grounded chain of names are reduced using 4-tuples that define a name in terms of a key. Eventually, 4-tuples of the former form are reduced to 4-tuples of the latter form. The validity date stored in the outcome of each reduction is the intersection of the validity dates of the 4-tuples being reduced.

Then the $\langle \text{name}, \text{key} \rangle$ bindings formed by the reduced 4-tuples are applied to resolve names in 5-tuples back to keys, again carrying validity dates through with intersection operations. This operation turns $\langle \text{authorization}, \text{name} \rangle$ 5-tuples into $\langle \text{authorization}, \text{key} \rangle$ tuples.

At this point, each 5-tuple represents a subject key (or threshold subject defined as a set of keys) with authorization to perform some set of actions on behalf of the issuer key. When two 5-tuples form a chain of delegation (the issuer of the second is the subject of the first, and the first tuple allows further delegation), the 5-tuples are reduced to a new tuple whose subject is the subject of the second tuple and whose issuer is the issuer of the first. The reduced tuple carries the intersection of the authorizations of the source tuples as its authorization, and the intersection of the validity dates of the source tuples as its validity dates. Finally, the reduced tuple carries the same delegation control bit as the second tuple did. Think of the delegation control bit as the coupling on the back of a boxcar; if the first tuple lacks it, the cars cannot couple; if the second tuple lacks it, the cars may couple, but the resulting "super-car" will also lack a rear coupling.

We return to SPKI in Section 7, where we apply our extended calculus to model SPKI.

5 The logic and semantics of restricted delegation

Having presented modal logic and the access control logic of Lampson *et al.* in Sections 2 and 3, we are ready to extend the logic. Figure 4 summarizes the symbols we use in the following sections. In Section 5.1, we extend the speaks-for relation with an argument restricting the delegation to a subset of statements. Then in Section 6.1, we extend the logic to incorporate SPKI-style linked local names.

5.1 The $\overset{T}{\Rightarrow}$ extension to the calculus of access control

In one paper, Lampson *et al.* mention in passing the idea of a qualified speaks-for operator [LABW92, page 272]. In this section, we intro-

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, $\wedge$, $\|$, and $\cdot N$
$\mathcal{N}$	N	The set of local names

Figure 4: *The symbols used to represent sets in this paper.*

duce our **regarding** operator, which formalizes the notion of the restricted speaks-for operator. It is written $B \overset{T}{\Rightarrow} A$, and read "B speaks for 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 **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 Section 3.5, where to delegate authority to a restricted subset of her resources, Alice had to define a role and install that role in the ACLs of each resource to be shared.

Restricted speaks-for is transitive:

$$\vdash (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{B}) \wedge (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}) \quad \text{(Axiom 22)}$$

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

$$\vdash (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \wedge \mathcal{C}) \overset{T}{\Rightarrow} (\mathcal{A} \wedge \mathcal{C}) \quad \text{(Axiom 23)}$$

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

$$\vdash \mathcal{C} \overset{T}{\Rightarrow} (\mathcal{A} \wedge \mathcal{B}) \equiv (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}) \wedge (\mathcal{C} \overset{T}{\Rightarrow} \mathcal{B}) \quad \text{(Axiom 24)}$$

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

$$\vdash (\mathcal{B} \overset{\mathcal{U}}{\Rightarrow} \mathcal{A}) \equiv (\mathcal{B} \Rightarrow \mathcal{A}) \quad \text{(Axiom 25)}$$

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

$$\begin{aligned} &\forall T' \subseteq T, \\ &\qquad \vdash (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{T'}{\Rightarrow} \mathcal{A}) \quad \text{(Axiom 26)} \end{aligned}$$

Using Axiom 26, a chain of delegations can be collapsed to a single delegation from the head principal in the chain to the tail, whose restriction set allows the intersection of the restriction sets of each of the original delegations.

$$\vdash (\mathcal{C} \overset{S}{\Rightarrow} \mathcal{B}) \wedge (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{C} \overset{S \cap T}{\Rightarrow} \mathcal{A}) \quad \text{(Theorem 27)}$$

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 8.

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 is not the case.

$$(B \overset{S}{\Rightarrow} A) \wedge (B \overset{T}{\Rightarrow} A) \not\supset B \overset{S \cup T}{\Rightarrow} A \quad \text{(Result 28)}$$

In Section 5.1.2, we describe a relation weaker than $\overset{T}{\Rightarrow}$ for which the positive claim holds.

Lampson's quoting operator on principals ($|$) is monotonic in both arguments over $\Rightarrow$. Quoting is still monotonic over $\overset{T}{\Rightarrow}$ in its left argument:

$$\vdash \left(\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}\right) \supset \mathcal{C}|\mathcal{B} \overset{T}{\Rightarrow} \mathcal{C}|\mathcal{A} \qquad \text{(Axiom 29)}$$

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

$$\left(\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}\right) \not\supset \mathcal{B}|\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}|\mathcal{C} \qquad \text{(Result 30)}$$

This result appears to limit the usefulness of quoting. The same counterexample that shows Result 30 shows the same property for the weaker speaks-for relation defined in Section 5.1.2; so it seems that the notion of quoting simply does not mix easily with restricted delegation.

We can, however, propagate 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 $\mathcal{C}\,\mathbf{says}$: $T \subseteq TC$, and if $\sigma \in TC$, then $(\mathcal{C}\,\mathbf{says}\,\sigma) \in TC$. Now $(T^*)C$ is the modal closure applied to the propositional closure of some original set T. With these definitions, we can show:

$$\vdash \left(\mathcal{B} \overset{(T^*)C}{\Rightarrow} \mathcal{A}\right) \supset \left(\mathcal{B}|\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}|\mathcal{C}\right) \qquad \text{(Axiom 31)}$$

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 $\mathcal{C}$ that it is willing to believe $\mathcal{B}$ about. It seems awkward, but it is a useful result. Why? Because any possible-worlds semantics wherein $(\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B}|\mathcal{C}' \overset{T}{\Rightarrow} \mathcal{A}|\mathcal{C}')$ for *all* principals $\mathcal{C}'$ must depend on every other principal relation. The introduction of malicious principals with cleverly-chosen relations into such a system can effectively expand T until $T = \mathcal{U}$.

5.1.1 Semantics of $\overset{T}{\Rightarrow}$

The semantic definition of $\overset{T}{\Rightarrow}$ 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 $B \overset{T}{\Rightarrow} A$ in the original model. The model $\overline{M}$ is a projection of M that only preserves information about statements in T.

To do this, we begin by defining an equivalence relation $\cong_T$: $W \times W$ that relates two worlds whenever they agree on all statements in T:

$$w \cong_T w' \ \textbf{iff} \ \left(\forall \sigma \in T, w \in \mathcal{E}(\sigma) \ \textbf{iff} \ w' \in \mathcal{E}(\sigma)\right) \qquad \text{(Definition 32)}$$

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

$$\phi_T(w) = \phi_T(w') \ \textbf{iff} \ w \cong_T w' \qquad \text{(Definition 33)}$$

We give a construction of $\phi_T(w)$ in Appendix Section A.1.

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

$$\phi_T^w(S_w) = \{\overline{w} \mid \exists w \in R,\ \overline{w} = \phi_T(w)\} \qquad \text{(Definition 34)}$$

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 our first semantic definition of restricted delegation:

$$\mathcal{E}(\mathcal{B} \overset{T}{\Rightarrow} A) = \begin{cases} W & \text{if } \forall w_0 \begin{pmatrix} \phi_T^w(\mathcal{R}(\mathcal{A})(w_0)) \subseteq \\ \phi_T^w(\mathcal{R}(\mathcal{B})(w_0)) \end{pmatrix} \\ \varnothing & \text{otherwise} \end{cases} \qquad \text{(Definition 35)}$$

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

$$\phi_T^+(R) = \left\{ \langle w_0, w_1' \rangle \,\middle|\, \exists w_1 \cong_T w_1',\ \langle w_0, w_1 \rangle \in R \right\} \qquad \text{(Definition 36)}$$

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 $\overset{T}{\Rightarrow}$:

$$\mathcal{E}(\mathcal{B} \overset{T}{\Rightarrow} A) = \begin{cases} W & \text{if } \mathcal{R}(\mathcal{A}) \subseteq \phi_T^+(\mathcal{R}(\mathcal{B})) \\ \varnothing & \text{otherwise} \end{cases} \qquad \text{(Definition 37)}$$

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 saves us work in Section 6. The equivalence is shown in Appendix A.2.

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 regards 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 knows everything B knows about statements in T. This intuition is exactly what we want for restricted delegation.

What happens if we take alternative definitions for restricted delegation? We explore one tempting but undesirable alternative in Appendix A.3. Presently, in Sections 5.1.2 and 5.1.3, we explore two intriguing possibilities.

5.1.2 The $\overset{T}{\rightarrow}$ relation

Abadi mentions a weaker version of the speaks-for operator, $B \rightarrow A$,[6] that is true exactly when $B\,\textbf{says}\,\sigma \supset A\,\textbf{says}\,\sigma$. But A may have some different reason than B to say σ. Semantically, A may consider a totally different set of worlds possible; it just happens that σ is still true in those worlds. For that reason, $B \rightarrow A$ is a weaker relation than $B \Rightarrow A$. The latter requires a special (subset) relationship to appear in the model.

We can define a semantics for these operators.

$$\mathcal{E}(B \rightarrow A) = \left\{ w \,\middle|\, \begin{array}{l} \forall \sigma \in \Sigma^*, \\ \mathcal{R}(B)(w) \subseteq \mathcal{E}(\sigma) \supset \mathcal{R}(A)(w) \subseteq \mathcal{E}(\sigma) \end{array} \right\}$$

$$\mathcal{E}(B \overset{T}{\rightarrow} A) = \left\{ w \,\middle|\, \begin{array}{l} \forall \sigma \in T, \\ \mathcal{R}(B)(w) \subseteq \mathcal{E}(\sigma) \supset \mathcal{R}(A)(w) \subseteq \mathcal{E}(\sigma) \end{array} \right\}$$

These are not particularly exciting definitions; they simply state just what the axiom says:

$$(B \rightarrow A) = (B\,\textbf{says}\,\sigma \supset A\,\textbf{says}\,\sigma)$$

It is obvious that weak restricted speaks-for degenerates into Abadi's unrestricted → operator, since $\mathcal{U} = \Sigma^*$:

$$B \overset{\mathcal{U}}{\rightarrow} A \equiv B \rightarrow A$$

5.1.3 The $\overset{T}{\Rrightarrow}$ relation

Having witnessed a weaker relation →, one may wonder why Abadi *et al.* preferred a definition of speaks-for (⇒) that was stronger than it needed to be. The intuition seems to be that the stronger semantics captures the fact that A understands B's *reasons* for believing various statements. Our $\overset{T}{\Rrightarrow}$ is "strong" in the same sense; it degenerates to ⇒ when $T = \mathcal{U}$.

Our first attempt to build a strong speaks-for relation, however, ended up too strong. We call the definition below our "*mighty*" speaks-for:

$$\mathcal{E}(B \overset{T}{\Rrightarrow} A) = \left\{ w \,\middle|\, \forall \sigma,\ \mathcal{R}(A)(w) - \bigcap_{\sigma \in T} \mathcal{E}(\sigma) \subseteq \mathcal{R}(B)(w) \right\}$$

The idea here is that if a subset relationship restricts A to say everything B says, then permitting A's relation to grow a little permits A to not say some of the statements B says (those not in T). The definition above preserves the desired relationship: if $B\,\textbf{says}\,\sigma$, every world B can see has σ true; because A can only see those worlds

[6] Personally, we pronounce it "B weakly speaks for A."

and other worlds where σ is true, $A\,\mathbf{says}\,\sigma$. So A has no "reasons" (edges to possible worlds) to not say σ that B does not also have.

The $\overset{T}{\Rrightarrow}$ relation seemed promising because it degenerated into $\Rightarrow$ when $T = \mathcal{U}$. It appears too strong, however. For example, if $T = \{s, \neg s\}$, then $\cap_T \mathcal{E}(s) = \varnothing$, so any $B \overset{T}{\Rrightarrow} A$ would imply $B \Rightarrow A$. The semantics are so strong that many interesting choices of T are not possible. In fact, we present $\overset{T}{\Rrightarrow}$ here precisely because it emphasizes the importance of having a satisfying semantics to lend intuitive meaning to the logic. We explore this issue further in Section 8.

5.1.4 Relationships between the relations

Both $\stackrel{T}{\Rightarrow}$ and $\overset{T}{\Rrightarrow}$ are strictly stronger than $\xrightarrow{T}$:

$$B \stackrel{T}{\Rightarrow} A \supset B \xrightarrow{T} A$$
$$B \xrightarrow{T} A \not\supset B \stackrel{T}{\Rightarrow} A$$
$$B \overset{T}{\Rrightarrow} A \supset B \xrightarrow{T} A$$
$$B \xrightarrow{T} A \not\supset B \overset{T}{\Rrightarrow} A$$

The $\stackrel{T}{\Rightarrow}$ relation is certainly not stronger than $\overset{T}{\Rrightarrow}$; it seems that it should be strictly weaker, but a corner case that prevents it from being so:

$$B \stackrel{T}{\Rightarrow} A \not\supset B \overset{T}{\Rrightarrow} A$$
$$B \overset{T}{\Rrightarrow} A \not\supset B \stackrel{T}{\Rightarrow} A$$

We establish each of these relationships in Appendix A.5, frequently using counterexamples to demonstrate what makes one relation weaker than another.

We introduced the weak and mighty speaks-for relations in the interest of completeness. Like Lampson *et al.*, we work primarily with the $\stackrel{T}{\Rightarrow}$ version of the relation.

5.1.5 Supplanting *controls*

Now that we have the restricted speaks-for relation, we can dispense with the special *controls* operator for building ACLs. Recall the special principal **1** from Section 3.5.1. Because it believes only truth, $(\mathbf{1}\,\mathbf{says}\,s) \supset s$ for all statements s. (Check it in the semantics!) That is, we have an implicit principal that controls all statements.

We can use the identity to replace every statement of the form A *controls* s with an equivalent one: $A \overset{\{s\}}{\Rightarrow}_{w_0} \mathbf{1}$. This statement ensures that if $A\,\mathbf{says}\,s$, then at the actual world w_0 of the model, $\mathbf{1}\,\mathbf{says}\,s$. Since the **1** relation only contains edges from a node to itself, this condition can only be satisfied by selecting an actual world w_0 where s is true.

Although this is completely equivalent to the special *controls* operator, it is more elegant in that all access control on s is uniformly encoded in $\stackrel{T}{\Rightarrow}$ statements. As with ACLs, the ground-level statements like $A \stackrel{T}{\Rightarrow} \mathbf{1}$ must be stored on the server that implements the operations in T, because **1** is not a cryptographic key that can sign authorization certificates. In contrast to ACLs, however, A can make further delegations $B \stackrel{T_2}{\Rightarrow} A$ that share part of A's permission, *even if A cannot change the ACLs* (ground-level statements). This ability to share resources through an unlimited number of delegations, each with the ability to restrict the delegated permissions, was our original motivation for developing this extension to Lampson's calculus.

5.1.6 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 $A\,\mathbf{as}\,R_2$ 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 [EFL$^+$99, p. 17].

With the speaks-for-regarding relation, A can introduce a new role R_2 for itself by allowing A **as** $R_2 \stackrel{T_2}{\Rightarrow} A$. In fact, roles are no longer necessary at all, but the **as** and **for** operator, 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 **as** R_{user} is weaker than A, and the way that B **as** R_{user} is weaker than B.

Let us build a model for a concrete example. Our model has two worlds, w_s and $w_{\overline{s}}$, where s is true and s is false, respectively. Assume that neither Alice nor Bob begin by believing s: $\neg A$ **says** s and $\neg B$ **says** s. Our model must have the relations:

$$\langle w_0, w_{\overline{s}} \rangle \in \mathcal{R}(A)$$
$$\langle w_0, w_{\overline{s}} \rangle \in \mathcal{R}(B)$$

(w_0 is a placeholder for whichever world is the actual world.) Now assume Alice's doppelganger (A **as** R_{user}) **says** s. To model this, we need $\mathcal{R}(A \textbf{ as } R_{\text{user}}) = \mathcal{R}(A) \circ \mathcal{R}(R_{\text{user}})$ to include only worlds where s is true. We want to preserve $\neg A$ **says** s, or else it would be the case that $(A \textbf{ as } R_{\text{user}}) \Rightarrow A$. That means we cannot change A's relation; so our only recourse is to use $\mathcal{R}(R_{\text{user}})$ to sever the edges leading to $w_{\overline{s}}$:

$$\langle w_{\overline{s}}, w_{\overline{s}} \rangle \notin \mathcal{R}(R_{\text{user}})$$

But because R_{user} is also used by Bob, we arrive at:

$$w_{\overline{s}} \notin (\mathcal{R}(B) \circ \mathcal{R}(R_{\text{user}}))(w_0)$$
$$w_{\overline{s}} \notin \mathcal{R}(B \textbf{ as } R_{\text{user}})(w_0)$$

Since B **as** R_{user} has no edges to the world $w_{\overline{s}}$ where s is false, our model supports the statement (B **as** R_{user}) **says** s. Using a common role has caused unexpected crosstalk between one principal and another.

5.1.7 Statement expiration

Lampson *et al.* treat expiration times casually in [LABW92, p. 270]: "Each premise has a *lifetime*, and the lifetime of the conclusion, and therefore of the credentials, is the lifetime of the shortest-lived premise." 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 Section 3.3 that the 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 takes place.[7] The restriction set T in a delegation $B \overset{T}{\Rightarrow} A$ can include restrictions on the times of the operations under consideration. So now every delegation may remain "valid" forever. After the last point in time permitted by T, the delegation becomes useless; any requested operation bears a timestamp preventing it from belonging to T. 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.

6 Names

Recall from Section 5.1.6 how roles are in a global "namespace," and that seems dangerous, because there can be some crosstalk between applications of the same role. SPKI names are promising, but they have the same property: identical names have different meaning depending on the "scope" in which they appear. To model names, we need to extend our logic and semantics.

We introduce to the logic a new set of primitive *names*, $\mathcal{N}$. We also extend principal expressions to include those of the form $P \cdot N$, where P is an arbitrary principal expression and $N \in \mathcal{N}$. $P \cdot N$ is read "P's N." Because $\cdot$ only accepts a principal as its left argument, there is no ambiguity in the order of operations; $P \cdot N_1 \cdot N_2$ can only be parenthesized $(P \cdot N_1) \cdot N_2$.

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

6.1 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:

$$\forall \text{ principals } \mathcal{A}, \text{ names } N :$$
$$(\mathcal{A}\,\textsf{\textbf{says}}\,(\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A} \cdot N)) \supset (\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A} \cdot N)$$

We do not take this statement as an axiom for the same reason that Abadi and Lampson do not accept the handoff axiom [LABW92, p. 715], [ABLP93, p. 273]. In particular, our semantics does not support it. Instead, as with the handoff axiom, we assume appropriate instances of it are assumed by the implementation.

Monotonicity. Second, 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.

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

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

$$(B \Rightarrow A \cdot N_{\text{barber}}) \supset$$
$$B \cdot N_{\text{butcher}} \Rightarrow A \cdot N_{\text{barber}} \cdot N_{\text{butcher}}$$

We take as our axiom a version of the rule generalized to $\overset{T}{\Rightarrow}$.

$$\forall \text{ principals } \mathcal{A}, \mathcal{B}, \text{ names } N,$$
$$\text{and sets of statements } T :$$
$$(\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \cdot N \overset{T}{\Rightarrow} \mathcal{A} \cdot N) \qquad \text{(Axiom 38)}$$

Distributivity. We combine the following pair of results

$$(\mathcal{A} \wedge \mathcal{B}) \cdot N \Rightarrow (\mathcal{A} \cdot N) \wedge (\mathcal{B} \cdot N) \quad \text{(Theorem 39)}$$
$$(\mathcal{A} \cdot N) \wedge (\mathcal{B} \cdot N) \Rightarrow (\mathcal{A} \wedge \mathcal{B}) \cdot N \qquad \text{(Axiom 40)}$$

to show that names distribute over principal conjunction:

$$(\mathcal{A} \wedge \mathcal{B}) \cdot N = (\mathcal{A} \cdot N) \wedge (\mathcal{B} \cdot N) \quad \text{(Theorem 41)}$$

Here is a motivating example: If Alice has two doctors Ed (E) and Fred (F), and Bob visits doctors Fred and George (G), then who is "(Alice and Bob)'s doctor?" Fred is the only person who serves as both people's doctor.

No quoting axiom. The principal $(A|B) \cdot N$ can be written, but we have yet to find a meaningful intuitive interpretation for it. $(A|B) \cdot N$ bears no obvious relation to $(A \cdot N)|(B \cdot N)$, for example. We allow the principal in our logic, but we have 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 first attempted to model name application ($\cdot$) with the quoting operator ($|$), because quoting satisfies Axiom 38, which led us to consider using roles to model names. Roles are also idempotent, however, which is undesirable. Surely the principal "Alice's barber's barber" will not always be the same as "Alice's barber."

It may be the case, though, that the application of a name can become idempotent. Take the example in Figure 5. In this example, let the symbol N stand for the name "barber." The upper solid left arrow represents an explicit statement made by Alice: $A\,\textsf{\textbf{says}}\,B \Rightarrow (A \cdot N)$; that is, Bob may serve as Alice's barber, and do anything "Alice's barber" is allowed to do. Similarly, the other solid left arrow represents Bob delegating Charlie as his barber. It turns out Charlie and Bob work in the same barber shop and cut each other's hair. The swooping solid arrow on the right represents Charlie delegating the responsibility of "Charlie's barber" to Bob. So $A \cdot N$ is "Alice's barber," and is controlled by her barber Bob. $A \cdot N \cdot N$ is "Alice's barber's barber," and is controlled by her barber's barber Charlie. Bob also has some control over "Alice's barber's barber," since he is free to change his barber from Charlie to another.

An interesting thing happens at the next level of name application. The literal name "Alice's barber's barber's barber," who we know as Bob, is actually *equal* to "Alice's barber's barber." It

$$\begin{tikzcd}
A & & \\
A\cdot N & B \arrow[l, Rightarrow] \arrow[ddr, Rightarrow, bend left] & \\
A\cdot N\cdot N & B\cdot N \arrow[l, dotted] & C \arrow[l, Rightarrow] \\
A\cdot N\cdot N\cdot N \arrow[u, dotted, bend left] & B\cdot N\cdot N \arrow[l, dotted] & C\cdot N \arrow[l, dotted]
\end{tikzcd}$$

Figure 5: *An example that shows when inherited names can become idempotent. Each arrow represents a speaks-for relationship; the text explains each arrow in more detail.*

is not that Bob becomes equal to Charlie, but that $\cdot N$ has become idempotent. In our case, both Charlie and Bob have control over $A\cdot N\cdot N$, so any further application of $\cdot N$ introduces no new restrictions on the resulting principal. The derived principal is equal to the parent principal. This conclusion is both intuitive, and valid in the semantics we present next, but cannot be proven using the logic.[8]

6.2 The semantics of names

We mentioned above that 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 5.1.6.

Instead, we model names as follows. First, we add a new element 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 set of relations $\mathcal{R}$. 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 we 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. Our definition, however, would end up circular (at requirement II) 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:

$$\begin{aligned}
\mathcal{R}(\mathcal{A}\cdot N) \subseteq & \phi_T^+(\mathcal{R}(\mathcal{B}\cdot N)) && \text{(I)} \\
& && (\forall \mathcal{B}, T : \mathcal{R}(\mathcal{A}) \subseteq \phi_T^+(\mathcal{R}(\mathcal{B}))) \\
\mathcal{R}(\mathcal{A}\cdot N) \subseteq & K(\mathcal{A}, N) && \text{(II)} \\
& && (\text{when } \mathcal{A} \in P) \\
\mathcal{R}(\mathcal{A}\cdot N) \subseteq & \mathcal{R}(\mathcal{B}\cdot N) \cup \mathcal{R}(\mathcal{C}\cdot N) && \text{(III)} \\
& && (\text{when } \mathcal{A} = \mathcal{B} \wedge \mathcal{C}) \\
& && \text{(Definition 42)}
\end{aligned}$$

Requirement I supports Axiom 38. 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 would do so by modifying $K(\mathcal{A}, N)$. Requirement III only applies to principal expressions that are conjunctions, and justifies Theorem 41. There is no question some such largest relation exists; since each requirement is a subset relation, at least the empty set satisfies all four.

[8] We accept incompleteness. Abadi mentioned that the original logic was incomplete. The $\overset{T}{\Rightarrow}$ relation certainly does not help matters, considering that completeness would involve introducing finite-set mathematics to the logic.

6.2.1 Abadi's semantics for linked local namespaces

Abadi gives an alternate logic and semantics for SPKI-style linked local namespaces [Aba98]. (He refers to SDSI, from which SPKI 2.0 derives.) Ours differs in three interesting ways.

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 name that ends in !!) 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:

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

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

Abadi's semantics maps each unqualified name to a single relation, and models name application as quoting (composition). The single relation modeling a name can introduce crosstalk between otherwise unconnected principals. Recall the example from Section 5.1.6. 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 (if $B|N \overset{T}{\Rightarrow}$, Charlie can do the things in T), but not everything. Enter Alice, who is not only omniscient ($A = \mathbf{1}$), but serves as her own doctor ($A \Rightarrow A|N$). The model requires that $\mathcal{R}(\mathbf{1}) \circ \mathcal{R}(N) \subseteq \mathcal{R}(\mathbf{1})$. At worst, $\mathcal{R}(N) = \mathcal{R}(\mathit{identity})$, causing $B|N = B$, enabling Charlie's doctor to make investment decisions on Charlie's behalf. At best, $\mathcal{R}(N) \subset \mathcal{R}(\mathit{identity})$, 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. We must admit that our semantics for names is at best opaque. Using an existential definition like "largest set satisfying the requirements" is not illuminating. We feel it is better than the alternative, though.

7 Modeling SPKI

Lampson's original calculus for access control is useful because its principals are general enough to model several parts of a computing system, from users to trusted servers to communications channels. Our addition of the ability to restrict authority with any delegation provides the needed extra power to make the calculus useful across administrative domains. Perhaps the most convincing evidence of this power is how well our extended calculus can model SPKI, an access-control system designed to span administrative domains.

Recall that the Simple Public Key Infrastructure (SPKI), which we reviewed in Section 4, principally provides access control, rather than identity authentication. In that sense, it shows its Lampson heritage. To formally model SPKI with our extension to Lampson's original calculus, we first give a construction that models the delegation-control bit.

7.1 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. In spite of our opinion, we present an 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 questionable axiom, undermining the simplicity of the semantics.

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

$$\vdash A_u \, \textbf{says} \, s \supset A_b \, \textbf{says} \, s \qquad \text{(Assumption 43)}$$

Certificates issued by 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).

7.2 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 5.1.7). 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 (in SPKI, a name principal can never utter a statement directly, only a key principal can). We find it curious, however, that SPKI name-binding certificates omit the authorization field. Why not allow a principal to say the following?

$$I_u \, \textbf{says} \, (S_{1b} \Rightarrow I_b \cdot N_{\text{barber}})$$
$$I_u \, \textbf{says} \, (S_{2b} \overset{\{\mathit{shampoo}\}}{\Rightarrow} I_b \cdot N_{\text{barber}})$$

In SPKI, such a refinement would require the user represented by I_u to create a subsidiary key, bind it to $I_b \cdot N_{\text{barber}}$, and then make the restricted authorizations to the subsidiary key.

7.3 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 gave a logic and semantics for linked local namespaces in Section 6. 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:

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

7.4 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 list $\binom{n}{k}$ 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:

$$\begin{aligned} P_1 \wedge P_2 &\Rightarrow A \\ P_1 \wedge P_3 &\Rightarrow A \\ P_2 \wedge P_3 &\Rightarrow A \end{aligned}$$

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

7.5 S-expressions

S-expressions, as used in authorization fields in SPKI, merely represent sets of primitive statements. Therefore, we simply model them using mathematical sets. We use the fact that neither S-expressions nor validity date fields can represent a set containing a negated primitive statement in our analysis in Section 8.

7.6 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 same key that is responsible for the request, the tuple's issuer must represent the server providing the requested service, and the specific request must belong to the authorization field of the reduced tuple.

It is clear that tuple reduction is sound with respect to our extended logic. When 5- and 4-tuples are encoded in the logic as shown in Sections 7.2 and 7.3, tuple-reduction simply constructs a proof from several applications of Theorem 27.

SPKI's decision procedure is not complete with respect to the logic, because many statements in the logic simply cannot be expressed as a SPKI certificate. For example, the issuer of a SPKI certificate cannot be a conjunct principal, and quoting principals are not used anywhere in SPKI.

7.7 Validity conditions

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

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

to mean that 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

$$R\,\textsf{says}\,I \overset{T_V}{\Rightarrow} R$$

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

$$\begin{gathered} I\,\textsf{says}\ \mathbf{0} \Rightarrow I|H_1 \\ I\,\textsf{says}\ \mathbf{0} \Rightarrow I|H_2 \\ \vdots \end{gathered}$$

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

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

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

Negative lists (CRLs) can be handled similarly; an implementation examining a revocation list would conclude $I\,\textsf{says}\,\mathbf{0} \Rightarrow I|H_1$ 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 $E\,\textsf{says}\,s$, and sends it to the revalidator R. R replies with a statement meant to be interpreted

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

Now both B_1 and $E \overset{\{s\}}{\Rightarrow} R|H_1$ 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.

7.8 What is not in SPKI

Conjunct principals ($A \wedge B$) are not first-class entities in SPKI, although they can appear as threshold subjects. One consequence is that SPKI does not have a rule that exploits Theorem 41.

Quoting principals are also missing from SPKI; Lampson's paper gives nice examples showing how quoting can help a multiplexed host or communications channel differentiate when it is working on behalf of one client rather than another [LABW92, Sections 4.3, 6.1, 6.2, and 7.1]. Without quoting, such a host has permission to make statements for either client, so it must perform an access-control check in advance of relaying a client's statement. The situation is analogous to a root POP server that duplicates the permissions checks of the kernel rather than changing its UID and letting the kernel verify the permissions on the system call itself. Quoting lets the multiplexed host defer the complete access-control decision to the final server verifying the proof. The result is a smaller trusted computing base and improved auditability.

8 Consequences of Restriction

When $C \overset{T}{\Rightarrow} B$ and $B \overset{V}{\Rightarrow} A$, we conclude $C \overset{T\cap V}{\Rightarrow} A$; this natural idea of transitive restriction appears in SPKI's access control decision procedure as well. When $B \overset{T}{\Rightarrow} A$ and $B \overset{V}{\Rightarrow} A$, though, we conclude $B \overset{T\cup V}{\Rightarrow} A$. This conclusion is certainly true for $\overset{T\cup V}{\Rightarrow}$ in SPKI, for if B makes a statement $s \in T\cup V$, one certificate or the other allows B to prove that it is authorized on s. SPKI, however, has no notion of representing the union of restriction sets; Certificate Result Certificates can be used to summarize the proof of intersected restrictions, but not of unioned ones. This limitation is because SPKI's representation of restriction sets (each an implicit intersection of an S-expressions and a validity date range) lends itself to intersection operations but not to unions.

The consequences of such extension operations are worth considering. For example,

$$(\mathcal{B} \overset{\{\sigma,\tau\}}{\rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{\{\sigma\wedge\tau\}}{\rightarrow} \mathcal{A}) \qquad \text{(Axiom 44)}$$

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.

If we employ our projected version of Abadi's speaks-for semantics, not only does

$$(\mathcal{B} \overset{\{\sigma,\tau\}}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{\{\sigma\wedge\tau\}}{\Rightarrow} \mathcal{A}) \qquad \text{(Axiom 45)}$$

hold, but also:

$$(\mathcal{B} \overset{\{\sigma\}}{\Rightarrow} \mathcal{A}) \supset (\mathcal{B} \overset{\{\neg\sigma\}}{\Rightarrow} \mathcal{A}) \qquad \text{(Axiom 46)}$$

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 $B \overset{\{s\}}{\Rightarrow} A$ is explicitly granted, B can also cause A to say the negation of s.

Perhaps scarier still is that

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

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

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

With these semantics, one might fear that no restriction is actually meaningful. How might we escape it? One option is to abandon the **K** axiom ($A\,\mathbf{believes}\,s \wedge A\,\mathbf{believes}\,(s \supset t) \supset A\,\mathbf{believes}\,t$), so that principals no longer believe every consequence of their beliefs. This option seems 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 $B \overset{T}{\rightarrow} A$ relation instead of $B \overset{T}{\Rightarrow} A$ 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 (by ignoring any such statements).

A conclusion is 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 such as that presented here must be wary of this hazard.

9 Snowflake and the $\overset{T}{\Rightarrow}$ relation

We are building a prototype distributed system called Snowflake to learn about how to span administrative domains. In Snowflake, we are using the extended calculus to represent authorizations across administrative domains.

While the extended calculus can model SPKI, it can also model more general systems. Specifically, SPKI is restricted in that every authorization delegation must occur between two cryptographic keys. Cryptography can be expensive, however, and within a trusted computing base, it is unnecessary. In Snowflake, communications between "kernels" (in our prototype, Java virtual machines) use cryptographic certificates equivalent to SPKI certificates. Within a kernel, however, there may be many principals to represent, and no reason to pay the computational price to represent them with cryptographic keys. Similarly, the delegations among those principals need not be represented with signed certificates. By leaning on the broader calculus, we can build a kernel with faster but equally sound internal access control that resorts to cryptography for "long-haul" communication.

10 Related Work

Neuman's proxies are tokens that confer a restricted subset of a principal's rights upon some other principal [Neu93]. Such *restricted proxies* can be represented as statements of the form $A\,\mathsf{says}\,B \overset{T}{\Rightarrow} A$, with T the set of restrictions. One goal of his system is to allow authorization decisions and accounting to be performed by a remote server. He describes how proxies can be used by object servers to delegate authorization responsibility to a third party. He also describes a scheme for accounting wherein proxies represent checks, endorsements, and certified checks. The check proxies allow object servers to transfer currency from the client to account for services rendered. Regarding Lampson's calculus, Neuman observes that "the creation of a new role is cumbersome when delegating on the fly or when granting access to individual objects" (page 288). Our work is motivated by the same sentiment, but we remove that hindrance while retaining the formal nature of Lampson's framework.

The PolicyMaker system of Blaze *et al.* is a framework for services that rely on cryptography [BFL96]. PolicyMaker assertions associate an authority structure (a set of public keys, which we would represent as principals) with a filter that defines the actions the principals are allowed to perform (analogous to the restriction sets of the $\overset{T}{\Rightarrow}$ relation). Such assertions are either "policies" or "certificates." The object server originates the former, and trusts them unconditionally. Certificates originate beyond the object server, and each is signed by its source, to establish its validity. Filters are general programs run in a resource-bounded safe language. They have access not only to the request, but to the "environment:" the current time, the name of the application, and the chain of keys and certificates that is being evaluated. We consider the entities in PolicyMaker to be unnecessarily specific. Lampson's calculus gives us a fundamental understanding of trust relationships separate from implementation choices such as cryptography.

Massacci's work on role-based access control defines a semantics for an access-control logic very similar to that of Lampson *et al.* [Mas97]. It has semantic limitations that allow the use of decisions based on tableau methods. His application of tableau methods either provide proof of authentication or a counter model that shows that no proof exists.

Bertino *et al.* offer a model for temporal access control, from expiration times to periodic authorizations such as allowing access on Tuesdays [BBFS96]. Their model allows statements premised upon the negation of other statements, and so requires global knowledge of the system to make decisions. This precludes its use in distributed systems, particularly those that span administrative domains.

Gray and Syverson provide a logic for reasoning about security in systems that require mandatory access control [GS98]. They are concerned with verifying that a system meets the definition of "probabilistic noninterference:" a principal only knows what it is permitted to know. Their logic is focused on the verification of a single central system being accessed by a set of adversarial programs at different classification levels.

11 Summary

We describe an extension to the access-control calculus of Lampson *et al.* [LABW92], in which the new *speaks-for-regarding* operator $\overset{T}{\Rightarrow}$ allows a principal to share a restricted subset of its authority with another principal. The three main advantages of this extension are:

1. A principal may delegate a restricted part of its authority, even when the objects of concern do not have ACLs writable by the delegator. Restricted delegation shows its power when used in chains of delegation, where each principal may pass on only a subset of its privileges to the next without needing to edit ACLs or offer a restricted signing oracle to do so.

2. The original *controls* operator for specifying of ACLs is replaced with a collection of $\overset{T}{\Rightarrow}$ expressions in the basic calculus. What once were called ACLs are now just a particular case of restricted delegation: the first delegations in any chain. That means that any policy that could be encoded before in ACLs can now be encoded anywhere in a chain of delegations, making a system much more flexible and manageable.

3. Time restrictions on statements, such as expiration times, are directly encoded in the calculus.

The first advantage adds new expressiveness to the calculus, expressiveness that is critical to the viability of the calculus as a security model in a widely distributed system that spans administrative domains. Because the concept of restricted delegation is supported in a general calculus that models arbitrary kinds of principals, the security model is as suitable for use both locally on a single host and between hosts with no *a priori* administrative relationship. Systems without this crucial property preclude users from communicating due to mechanisms that make assumptions about the structure of administrative domains. Systems with this property can still build policies reflecting desired administrative relationships using restricted delegation; it is just that the mechanism of the system no longer dictates the structure of those administrative relationships.

We also applied the extended calculus to modeling SPKI and our own distributed system. We discussed some of the consequences exposed by our extended logic and its semantics; specifically, the fact that a logical system can present the illusion of restricted delegation when in fact the delegation restricts very little or not at all. SPKI narrowly escapes this fate, but related systems or proposals to extend SPKI should carefully consider this problem.

Acknowledgements

Thanks to John Lamping, who patiently helped Jon understand logical proof systems and semantic models. Thanks also Doug McIlroy, 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

[Aba98] M. Abadi. On SDSI's linked local name spaces. *Journal of Computer Security*, 6(1-2):3–21, 1998.

[ABLP93] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, September 1993.

[BBFS96] Elisa Bertino, Claudio Bettini, Elena Ferrari, and Pierangela Samarati. Supporting periodic authorizations and temporal reasoning in database access control. In *Proceedings of 22nd International Conference on Very Large Data Bases*, pages 472–483. Morgan Kaufmann, September 1996.

[BFL96] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, 1996.

[EFL+99] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylonen. SPKI certificate theory, October 1999. Internet RFC 2693.

[FHMV95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge.* MIT Press, 1995.

[Gol73] William Goldman. *The Princess Bride.* Ballantine, 1973.

[GS98] J. W. Gray III and P. F. Syverson. A logical approach to multilevel security of probabilistic systems. *Distributed Computing*, 11(2):73–90, 1998.

[HC96] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic.* Routledge, 1996.

[LABW92] 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.

[Mas97] F. Massacci. Reasoning about security: a logic and a decision method for role-based access control. In *Proceedings of the First International Joint Conference on Qualitative and Quantitative Practical Reasoning (ECSQARU-FAPR)*, pages 421–435, 1997.

[Neu93] B. Clifford Neuman. Proxy-based authorization and accounting for distributed systems. In *Proceedings of the 13th International Conference on Distributed Computing Systems (ICDCS)*, pages 283–291, May 1993.

[WABL94] Edward Wobber, Martín Abadi, Michael Burrows, and Butler Lampson. Authentication in the Taos operating system. *ACM Transactions on Computer Systems*, 12(1):3–32, February 1994.

A Proofs

A.1 Construction of ϕ_T

Definition Definition 33 presupposed the existence of a projection function ϕ_T. We construct such a function now, and show that it satisfies the definition. Let $\overline{W} = 2^T$; that is, worlds in $\overline{M}$ are subsets of T. Define

$$\phi_T(w) = \overline{w} \in \overline{W}$$
$$\text{where } (\sigma \in \overline{w}) \equiv (w \in \mathcal{E}(\sigma)) \ \forall \sigma \in T \qquad \text{(Definition 47)}$$

Necessity. Given $\phi_T(w) = \overline{w} = \phi_T(w')$, we know $\forall \sigma \in T$, $\sigma \in \overline{w}$ **iff** $w \in \mathcal{E}(\sigma)$, and likewise,

$\forall \sigma \in T,\ \sigma \in \overline{w}$ **iff** $w' \in \mathcal{E}(\sigma)$. Therefore $\forall s \in T,\ w \in \mathcal{E}(\sigma)$ **iff** $w' \in \mathcal{E}(\sigma)$, and we conclude $w \cong_T w'$.

Sufficiency. From the definition of $w \cong_T w'$, we know $\forall \sigma \in T,\ w \in \mathcal{E}(\sigma)$ **iff** $w' \in \mathcal{E}(\sigma)$. Let $\overline{w} = \{\sigma \in T | w \in \mathcal{E}(\sigma)\}$ and $\overline{w}' = \{\sigma \in T | w' \in \mathcal{E}(\sigma)\}$. From our hypothesis we know that the conditions on $\overline{w}$ and $\overline{w}'$ are the same, so $\phi_T(w) = \overline{w} = \overline{w}' = \phi_T(w')$.

In the following proofs, we generally use a bar ($\overline{w}$) to indicate a member of an equivalence class constructed as shown here.

A.2 Equivalence of ϕ_T^R and ϕ_T^+ definitions of $\overset{T}{\Rightarrow}$

We now justify our claim in Section 5.1.1 that Definition 35 and Definition 37 are equivalent.

Necessity. Assume $\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}$ holds according to Definition 35:

$$\forall w_0' \ \left(\phi_T^w(\mathcal{R}(\mathcal{A})(w_0')) \subseteq \phi_T^w(\mathcal{R}(\mathcal{B})(w_0'))\right)$$

For all $\langle w_0, w_1 \rangle$,

$$\begin{aligned}
\langle w_0, w_1 \rangle \in \mathcal{R}(\mathcal{A}) &\supset w_1 \in \mathcal{R}(\mathcal{A})(w_0) \\
&\supset \ \overline{w}_1 \in \phi_T^w(\mathcal{R}(\mathcal{A})(w_0)), \\
&\qquad\qquad \overline{w}_1 = \phi_T(w_1) \\
&\supset \overline{w}_1 \in \phi_T^w(\mathcal{R}(\mathcal{B})(w_0)) \\
&\qquad\qquad \text{(using the assumption)} \\
&\supset \ \exists w_1' \cong_T w_1, \\
&\qquad \langle w_0, w_1' \rangle \in \mathcal{R}(\mathcal{B})(w_0) \\
&\supset \langle w_0, w_1 \rangle \in \phi_T^+(\mathcal{R}(\mathcal{B}))
\end{aligned}$$

Sufficiency. Assume $\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A}$ holds according to Definition 37:

$$\mathcal{R}(\mathcal{A}) \subseteq \phi_T^+(\mathcal{R}(\mathcal{B}))$$

Given w_0 and $\overline{w}_1 \in \phi_T^w(\mathcal{R}(\mathcal{A})(w_0))$, we know that there is some $w_1 \in \mathcal{R}(\mathcal{A})(w_0)$, with $\overline{w}_1 = \phi_T(w_1)$. We rewrite the statement $\langle w_0, w_1 \rangle \in \mathcal{R}(\mathcal{A})$, and invoke the assumption to get $\langle w_0, w_1 \rangle \in \phi_T^+(\mathcal{R}(\mathcal{B}))$. Now we know there exists $\langle w_0, w_1' \rangle \in \mathcal{R}(\mathcal{B})$ with $w_1' \cong_T w_1$. Changing notation again, $w_1' \in \mathcal{R}(\mathcal{B})(w_0)$. Since $w_1' \cong_T w_1$, we know $\overline{w}_1 = \phi_T(w_1')$, and we may conclude $\overline{w}_1 \in \phi_T^w(\mathcal{R}(\mathcal{B})(w_0))$.

Together, the two implications show the equivalence.

A.3 An undesirable semantics for $\overset{T}{\Rightarrow}$

Notice that ϕ_T^+ projects only the destination world of each edge in a relation. Why do we not project both ends of the relation? Such a definition actually does not preserve our most basic intuition, that $B \overset{T}{\Rightarrow} A \supset B \overset{T}{\rightarrow} A$. In the model in Figure 6, the dotted ovals depict the equivalence classes under T; projecting both ends of the edges in $\mathcal{R}(A)$ gives $\{\langle T, \varnothing \rangle\}$, as does $\mathcal{R}(B)$. From world w_0, however, B **says** s but not A **says** s.

Given a relation $\langle w_0, w_1 \rangle$, then, the reason we only project w_1 is this: w_0 is affected by what statements are true at w_1; substituting other worlds equivalent with respect to T does no harm. Substituting other worlds for w_0, on the other hand, changes what statements we consider true at w_0.

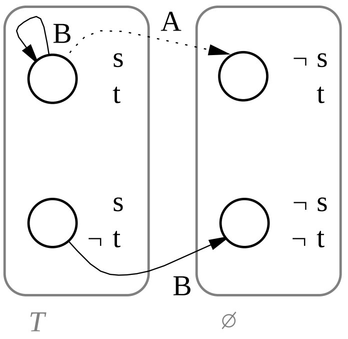


Figure 6: *In this example, $T = \{s\}$. Notice that $B \overset{T}{\not\Rightarrow} A$.*

A.4 Proof of soundness

In this section, we show that our extension to Lampson's calculus is still a sound axiomatization of the presented semantics. Like Lampson's original logic, ours is based on a conventional Kripke semantics of modal logic. The conventional proofs of soundness for Axiom 1, Rule 2, Axiom 3, and Rule 4 apply. Our extensions define $\mathcal{E}$ for a new formula $(\mathcal{B} \overset{T}{\Rightarrow} \mathcal{A})$ and $\mathcal{R}$ for a new principal $(\mathcal{A}{\cdot}N)$, but do not perturb Abadi's original semantics for the calculus for access control. Because those semantics do not depend on any particular structure in $\mathcal{E}$ or $\mathcal{R}$, the axioms of the calculus remain sound in our extended calculus.

Our present task is to show that the axioms of our extensions are sound.

Axiom 22. This axiom follows easily from Definition 35. For all w_0,

$$\begin{aligned} \phi^w_T(\mathcal{R}(\mathcal{A})(w_0)) &\subseteq \phi^w_T(\mathcal{R}(\mathcal{B})(w_0)) \\ &\subseteq \phi^w_T(\mathcal{R}(\mathcal{C})(w_0)) \qquad \square \end{aligned}$$

The following lemma shows that ϕ^+_T preserves the union operation. Let R_1 and R_2 be relations.

$$\begin{aligned} &\langle w_0, w_1\rangle \in \phi^+_T(R_1 \cup R_2) \\ &\quad \equiv \exists\, w'_1 \cong_T w_1,\ \langle w_0, w'_1\rangle \in R_1 \cup R_2 \\ &\quad \equiv \exists\, w'_1 \cong_T w_1, \\ &\qquad\quad \langle w_0, w'_1\rangle \in R_1 \vee \langle w_0, w'_1\rangle \in R_2 \\ &\quad \equiv \quad \exists\, w'_1 \cong_T w_1,\ \langle w_0, w'_1\rangle \in R_1 \\ &\qquad \vee \exists\, w'_1 \cong_T w_1,\ \langle w_0, w'_1\rangle \in R_2 \\ &\quad \equiv \langle w_0, w_1\rangle \in \phi^+_T(R_1) \vee \langle w_0, w_1\rangle \in \phi^+_T(R_2) \\ &\quad \equiv \langle w_0, w_1\rangle \in \phi^+_T(R_1) \cup \phi^+_T(R_2) \end{aligned}$$

From this equivalence we conclude

$$\phi^+_T(R_1 \cup R_2) = \phi^+_T(R_1) \cup \phi^+_T(R_2) \quad \text{(Lemma 48)}$$

Axiom 23. We assume the premise in terms of Definition 35:

$$\forall\, w'_0\ (\phi^w_T(\mathcal{R}(\mathcal{A})(w'_0)) \subseteq \phi^w_T(\mathcal{R}(\mathcal{B})(w'_0)))$$

We can readily reason for all w_0:

$$\begin{aligned} &\phi^w_T(\mathcal{R}(\mathcal{A} \wedge \mathcal{C})(w_0)) \\ &\quad = \phi^w_T((\mathcal{R}(\mathcal{A}) \cup \mathcal{R}(\mathcal{C}))(w_0)) \\ &\quad = \phi^w_T(\mathcal{R}(\mathcal{A})(w_0) \cup \mathcal{R}(\mathcal{C})(w_0)) \\ &\quad = \phi^w_T(\mathcal{R}(\mathcal{A})(w_0)) \cup \phi^w_T(\mathcal{R}(\mathcal{C})(w_0)) \\ &\quad \subseteq \phi^w_T(\mathcal{R}(\mathcal{B})) \cup \phi^w_T(\mathcal{R}(\mathcal{C})) \\ &\quad = \phi^w_T(\mathcal{R}(\mathcal{B})(w_0) \cup \mathcal{R}(\mathcal{C})(w_0)) \\ &\quad = \phi^w_T((\mathcal{R}(\mathcal{B}) \cup \mathcal{R}(\mathcal{C}))(w_0)) \\ &\quad = \phi^w_T(\mathcal{R}(\mathcal{B} \wedge \mathcal{C})(w_0)) \qquad \square \end{aligned}$$

Axiom 24. This axiom has a symmetric consequence, so we only show the first conjunct. For all worlds w_0,

$$\begin{aligned} \phi^w_T(\mathcal{R}(\mathcal{A})(w_0)) &\subseteq \phi^R_T((\mathcal{R}(\mathcal{A}) \cup \mathcal{R}(\mathcal{B}))(w_0)) \\ &\qquad\qquad \text{(Lemma 48)} \\ &\subseteq \phi^R_T(\mathcal{R}(\mathcal{C})(w_0)) \qquad \text{(premise)} \\ &\qquad\qquad \square \end{aligned}$$

We digress to point out that we may discard "identical worlds" from a model without loss of generality. That is, imagine we have a model $\mathcal{M}$ with two worlds w_1 and w_2 where $w_1 \in \mathcal{E}(\sigma)$ **iff** $w_2 \in \mathcal{E}(\sigma)$ for every formula $\sigma \in \Sigma^*$. The extra world w_2 appears in every $I(s)$ that w_1 appears in. Any edge in any relation ending in w_1 has a related edge ending in w_2 ($\langle w, w_1\rangle \in J(\mathcal{A}) \equiv \langle w, w_2\rangle \in J(\mathcal{A})$); likewise edges starting at w_1 have a related edge starting at w_2 in every relation. The same holds for the relations in the name interpretation function $K(\mathcal{A}, N)$. It is clear that the extension function $\mathcal{R}$, and hence $\mathcal{E}$, have the same overlap with respect to w_1 and w_2, so that $w_1 \in \mathcal{E}(\sigma) \equiv w_2 \in \mathcal{E}(\sigma)$.

Given this definition, we can build a model $\mathcal{M}' = \langle W', w'_0, I', J', K'\rangle$ that discards w_2:

$$\begin{aligned} W' &= W - \{w_2\} \\ w'_0 &= \begin{cases} w_1 & \text{if } w_0 \\ w_0 & \text{otherwise} \end{cases} \\ I'(s) &= I(s) - \{w_2\} \\ J'(\mathcal{A}) &= J(\mathcal{A}) - \{\langle w, w'\rangle | w = w_2 \vee w' = w_2\} \\ K'(\mathcal{A}, N) &= K(\mathcal{A}, N) \\ &\quad - \{\langle w, w'\rangle | w = w_2 \vee w' = w_2\} \end{aligned}$$

Happily, $\mathcal{M}'$ preserves every consequence of $\mathcal{M}$: $(\mathcal{M} \models \sigma) \equiv (\mathcal{M}' \models \sigma)$. Why? Whenever $w_0 \in \mathcal{E}(\sigma)$, $w'_0 \in \mathcal{E}'(\sigma)$, either for exactly the same reasons (when $w_0 \neq w_2$), or because $w_0 = w_2$, so $w_0 = w_2 \in \mathcal{E}(\sigma) \equiv w_1 \in \mathcal{E}(\sigma)$, and then $w'_0 \in \mathcal{E}'(s)$ for the same reasons that $w_1 \in \mathcal{E}(s)$.

Convinced that duplicate worlds do not alter the consequences of a model, we may now assume that no models contain identical worlds, without damaging our semantics. If we know $w_1 \neq w_2$, we can assume the existence of a formula σ with

$(w_1 \in \mathcal{E}(\sigma)) \not\equiv (w_2 \in \mathcal{E}(\sigma))$, and conclude that $w_1 \not\equiv_{\mathcal{U}} w_2$ (by Definition 32). Therefore, $\phi_{\mathcal{U}}$ is bijective:

$$w_1 \neq w_2 \supset \phi_{\mathcal{U}}(w_1) \neq \phi_{\mathcal{U}}(w_2)$$

By the definition of ϕ^+_T it is obvious that any relation $R \in \phi^+_T(R)$. But when $T = \mathcal{U}$, the converse is also true:

$$\begin{aligned}
&\langle w_0, w_1 \rangle \in \phi^+_{\mathcal{U}}(R) \\
&\quad \supset \exists\ w'_1 \text{ such that } \langle w_0, w'_1 \rangle \in R, \\
&\qquad \phi_{\mathcal{U}}(w'_1) = \phi_{\mathcal{U}}(w_1) \\
&\quad \supset w'_1 = w_1 \\
&\quad \supset \langle w_0, w_1 \rangle \in R
\end{aligned}$$

Now we have $\phi^+_{\mathcal{U}}(R) = R$.

Axiom 25. Expanding the definition of $B \overset{\mathcal{U}}{\Rightarrow} A$ and applying the previous result gives $\mathcal{R}(A) \subseteq \phi^+_{\mathcal{U}}(\mathcal{R}(B)) = \mathcal{R}(B)$, which satisfies the definition of $B \Rightarrow A$. □

Justifying axiom Axiom 26 requires two lemmas that relate representatives of equivalence classes under different projections.

First, a representative of a projection due to a small set has a "big brother" in any projection due to a superset, and the structure of the brothers is closely related:

$$\begin{aligned}
&\overline{w}'_1 \in \phi^w_{T'}(S_w),\ T' \subseteq T \\
&\supset \exists\, \overline{w}_1 \in \phi^w_T(S_w),\ \overline{w}'_1 = \overline{w}_1 \cap T' \quad \text{(Lemma 49)}
\end{aligned}$$

Proof. By the first premise, there is a $w_1 \in S_w$ where $\overline{w}'_1 = \phi_{T'}(w_1)$. From Definition 47 we know

$$(\sigma \in \overline{w}'_1) \equiv (w_1 \in \mathcal{E}(\sigma))\ \forall\, \sigma \in T' \qquad (1)$$

Let $\overline{w}_1 = \phi_T(w_1)$; since $w_1 \in S_w$, $\overline{w}_1 \in \phi^w_T(S_w)$. Having exhibited $\overline{w}_1$, we need only show $\overline{w}_1 \cap T = \overline{w}'_1$.

We again invoke Definition 47 to get

$$(\sigma \in \overline{w}_1) \equiv (w_1 \in \mathcal{E}(\sigma))\ \forall\, \sigma \in T \qquad (2)$$

First, $\sigma \in \overline{w}_1 \cap T'$ means both $\sigma \in T'$, and because $T' \subseteq T$, $\sigma \in T$. The latter allows us to use (2) to write $w_1 \in \mathcal{E}(\sigma)$, and then we invoke (1) to get $\sigma \in \overline{w}'_1$. Conversely, $\sigma \in \overline{w}'_1$ means $\sigma \in T'$ and hence $\sigma \in T$. We apply (1) to get $w_1 \in \mathcal{E}(\sigma)$, and apply (2) to get $\sigma \in \overline{w}_1$. Now we have shown $\overline{w}_1 \cap T' = \overline{w}'_1$, proving the lemma. □

The second lemma is approximately the converse of the first:

$$\begin{aligned}
&\overline{w}_1 \in \phi^w_T(S_w),\ \overline{w}'_1 = \overline{w}_1 \cap T'\ T' \subseteq T \\
&\supset \overline{w}'_1 \in \phi^w_{T'}(S_w) \qquad \text{(Lemma 50)}
\end{aligned}$$

Proof. The first premise, by Definition 34, implies the existence of a $w_1 \in R$, and Definition 47 lets us write

$$(\sigma \in \overline{w}_1) \equiv (w_1 \in \mathcal{E}(\sigma))\ \forall\, \sigma \in T \qquad (1)$$

For every $\sigma \in T'$, all of the following hold:

$$\begin{aligned}
\sigma &\in T && \text{(third premise)} \\
(\sigma \in \overline{w}_1) &\equiv (w_1 \in \mathcal{E}(\sigma)) && (1) \\
(\sigma \in \overline{w}_1 \cup T') &\equiv (w_1 \in \mathcal{E}(\sigma)) && \\
(\sigma \in \overline{w}'_1) &\equiv (w_1 \in \mathcal{E}(\sigma)) && \text{(second premise)}
\end{aligned}$$

This last result implies that $\overline{w}'_1 = \phi_{T'}(w_1)$, which is sufficient to prove the conclusion of the lemma. □

Axiom 26. We take as our hypothesis $\mathcal{M} \models B \overset{T}{\Rightarrow} A$, that is:

$$\phi^w_T(\mathcal{R}(\mathcal{A})(w_0)) \subseteq \phi^w_T(\mathcal{R}(\mathcal{B})(w_0))$$

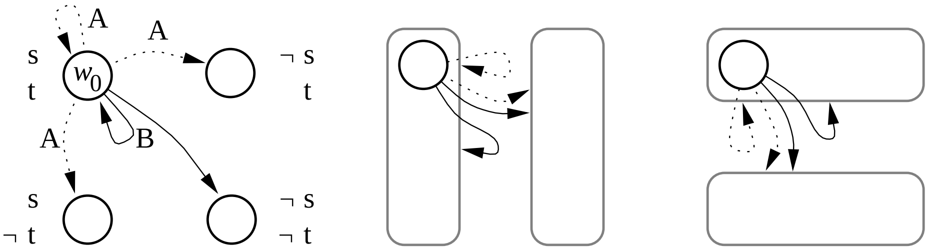
Given any world w_0 and sets $T' \subseteq T$, we assume $\overline{w}'_1 \in \phi^w_{T'}(\mathcal{R}(\mathcal{A})(w_0))$ and set out to prove $\overline{w}'_1 \in \phi^w_{T'}(\mathcal{R}(\mathcal{B})(w_0))$. By the assumption and Lemma 49, we know

$$\exists\, \overline{w}_1 \in \phi^w_T(\mathcal{R}(\mathcal{A})(w_0)),\ \overline{w}'_1 = \overline{w}_1 \cap T'$$

The hypothesis gives $\overline{w}_1 \in \phi^w_T(\mathcal{R}(\mathcal{B})(w_0))$, which satisfies the premise for Lemma 50. Hence we know $\overline{w}'_1 \in \phi^w_{T'}(\mathcal{R}(\mathcal{B})(w_0))$, and we have proven that

$$\forall\, w_0,\ (\phi^w_{T'}(\mathcal{R}(\mathcal{A})(w_0)) \subseteq \phi^w_{T'}(\mathcal{R}(\mathcal{B})(w_0))) \quad □$$

Theorem 27. Apply Axiom 26 twice to the premises to get two relations restricted by $S \cup T$, then apply Axiom 22 to collapse them into the relation in the conclusion. □



In this model, A's relation at w_0 is not a subset of B's.

Projected under $S = \{s\}$, however, the subset relation holds . . .

. . . as it does under $T = \{t\}$.

Figure 7: *A counterexample showing why two delegations for sets S and T do not imply a delegation for set $S \cup T$ (Result 28).*

Result 28. Figure 7 gives a counterexample that justifies the result. The diagram in the figure models $B \overset{S}{\Rightarrow} A$ and $B \overset{T}{\Rightarrow} A$. The statement $B \overset{S\cup T}{\Rightarrow}$, however, fails. Projecting the model under $S \cup T$ gives the original picture, since each world falls in a separate equivalence class. Notice that B says $\neg(s \wedge \neg t)$: that statement is true in both worlds B considers possible. But A does not believe it, since A can see the lower-left world, where the statement is false.

Why should this result be intuitive or desirable? Recall from Section 8 that the strength of $\overset{T}{\Rightarrow}$ means that a delegation regarding T may imply a delegation regarding a larger set T^* that includes formulas constructed from the members of T. In our example, B speaks for A regarding formulas composed exclusively with the primitive s or the primitive t, but not regarding formulas combining the two. The closure of the restriction set $S \cup T$ includes formulas such as $\neg(s \wedge \neg t)$.

Axiom 29. Assume the premise in terms of Definition 35:

$$\forall\, w_0' \; (\phi_T^w(\mathcal{R}(\mathcal{A})(w_0')) \subseteq \phi_T^w(\mathcal{R}(\mathcal{B})(w_0')))$$

Let $\overline{w}$ belong to $\phi_T^w(\mathcal{R}(\mathcal{C}|\mathcal{A})(w_0))$. The semantics for quoting gives $\overline{w} \in \phi_T^w((\mathcal{R}(\mathcal{C}) \circ \mathcal{R}(\mathcal{A}))(w_0))$. An edge only exists in a composition if we have w_1 and w_2 such that $\langle w_0, w_1\rangle \in \mathcal{R}(\mathcal{C})$ and $\langle w_1, w_2\rangle \in \mathcal{R}(\mathcal{A})$; Definition 34 guarantees that we have such w_1, w_2 with $\overline{w} = \phi_T(w_2)$.

Since $w_2 \in \mathcal{R}(\mathcal{A})(w_1)$, we can use the assumption to show the existence of $w_2' \in \mathcal{R}(\mathcal{B})(w_1)$ with $\phi_T(w_2') = \phi_T(w_2) = \overline{w}$. That means that $\overline{w} \in \phi_T^w(\mathcal{R}(\mathcal{B})(w_1))$, and hence $\overline{w} \in \phi_T^w((\mathcal{R}(\mathcal{C}) \circ \mathcal{R}(\mathcal{B}))(w_0))$. By the definition of quoting, we arrive at $\overline{w} \in \phi_T^w(\mathcal{R}(\mathcal{C}|\mathcal{B})(w_0))$, which proves the conclusion. □

Result 30. The model in Figure 8 is a counterexample for $T = \{s\}$ that shows the result. Notice that $B \overset{T}{\Rightarrow} A$: $\mathcal{R}(A)$'s only edge goes from w_0 to the equivalence class of worlds where s is true, and $\mathcal{R}(B)$ also has such an edge (the loop at w_0). When we compose the relations, however, we see that $B|C$ **says** s, but not $A|C$ **says** s. The equivalence classes of $\{C\,\textbf{says}\,s\}$ are different than the equivalence classes of $\{s\}$.

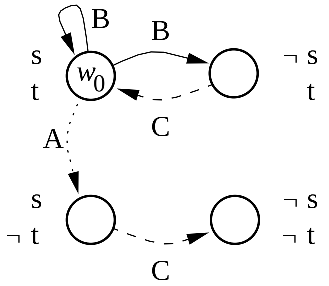


Figure 8: *A model that demonstrates Result 30.*

Axiom 31. Inductively applying Axiom 46 and Axiom 45 shows as a theorem that $B \overset{T}{\Rightarrow} A$ implies $B \overset{T^*}{\Rightarrow} A$. Therefore, we may immediately replace the premise of this axiom with $B \overset{((T^*)C)^*}{\Rightarrow} A$, which follows by the theorem from the original premise. Herein we omit the parentheses for the postfix set operators * and C, and simply write T^*C^*.

Hence we begin with the hypothesis that

$$\mathcal{R}(\mathcal{A}) \subseteq \phi^+_{T^*C^*}(\mathcal{R}(\mathcal{B}))$$

We are given some $w_0 \in W$ and the existence of $\overline{w}_2 \in \phi^w_T(\mathcal{R}(\mathcal{A}|\mathcal{C})(w_0))$. The set can be rewritten $\phi^w_T((\mathcal{R}(\mathcal{A}) \circ \mathcal{R}(\mathcal{C}))(w_0))$, so we know that there exist w_1 and w_2, where

$$\begin{aligned} \langle w_0, w_1\rangle &\in \mathcal{R}(\mathcal{A}) \\ \langle w_1, w_2\rangle &\in \mathcal{R}(\mathcal{C}) \\ \overline{w}_2 &= \phi_{T^*C^*}(w_2) \end{aligned}$$

The last expression means that for all $\sigma \in T$, $\sigma \in \overline{w}_2$ if and only if $w_1 \in \mathcal{E}(\sigma)$.

Define the formula

$$\tau_2 = \bigwedge_{\sigma\in T} \begin{cases} \sigma & \text{if } \sigma \in \overline{w}_2 \\ \neg\sigma & \text{otherwise} \end{cases}$$

Intuitively, τ_2 is true at precisely those worlds that map to $\overline{w}_2$ under ϕ_T. We have constructed τ_2 such that $w_2 \in \mathcal{E}(\tau_2)$.

Since $\langle w_1, w_2\rangle \in \mathcal{R}(\mathcal{C})$, we know $\mathcal{R}(\mathcal{C}) \not\subseteq \mathcal{E}(\neg\tau_2)$, and therefore $w_1 \notin \mathcal{E}(\mathcal{C}\,\mathbf{says}\,\neg\tau_2)$, and finally $w_1 \in \mathcal{E}(\neg\mathcal{C}\,\mathbf{says}\,\neg\tau_2)$. The propositional closure of T ensures that each conjunct of τ_2, and thus τ_2 itself and $\neg\tau_2$, appear in T^*. The modal closure over "$\mathcal{C}$ **says**" ensures that $(\mathcal{C}\,\mathbf{says}\,\neg\tau_2) \in T^*C$, and therefore $(\neg\mathcal{C}\,\mathbf{says}\,\neg\tau_2) \in T^*C^*$.

Now we may employ the hypothesis to show that there exists a $w'_1 \in \mathcal{R}(\mathcal{B})(w_0)$ with $w'_1 \cong_{T^*C^*} w_1$. It follows that:

$$\begin{aligned} w'_1 &\in \mathcal{E}(\neg\mathcal{C}\,\mathbf{says}\,\neg\tau_2) \\ &= W - \mathcal{E}(\mathcal{C}\,\mathbf{says}\,\neg\tau_2) \\ &= W - \{w|\mathcal{R}(\mathcal{C})(w) \subseteq \mathcal{E}(\neg\tau_2)\} \\ &= \{w|\mathcal{R}(\mathcal{C})(w) \not\subseteq \mathcal{E}(\neg\tau_2)\} \\ &= \{w|\exists\, w'_2 \in \mathcal{R}(\mathcal{C})(w),\ w'_2 \notin \mathcal{E}(\neg\tau_2)\} \\ &= \{w|\exists\, w'_2 \in \mathcal{R}(\mathcal{C})(w),\ w'_2 \in \mathcal{E}(\tau_2)\} \end{aligned}$$

That is, we know there is a $w'_2 \in \mathcal{E}(\tau_2)$, with $\langle w'_1, w'_2\rangle \in \mathcal{R}(\mathcal{C})$.

With both $\langle w_0, w'_1\rangle \in \mathcal{R}(\mathcal{B})$ and $\langle w'_1, w'_2\rangle \in \mathcal{R}(\mathcal{C})$, we have $\langle w_0, w'_2\rangle \in \mathcal{R}(\mathcal{B})\circ\mathcal{R}(\mathcal{C}) = \mathcal{R}(\mathcal{B}|\mathcal{C})$. From the definition of τ_2, we know that w'_2 is in $\mathcal{E}(\sigma)$ exactly when $\sigma \in \overline{w}_2$ for all $\sigma \in T$, so $\overline{w}_2 = \phi_T(w'_2)$. We have shown that $\overline{w}_2 \in \phi^w_T(\mathcal{R}(\mathcal{B}|\mathcal{C})(w_0))$, and therefore that given the hypothesis, the model supports $\mathcal{B}|\mathcal{C} \overset{T}{\Rightarrow} \mathcal{A}|\mathcal{C}$. □

Axiom 38. This axiom follows from our current brute-force semantics for names. Assume the premise in terms of Definition 37:

$$\mathcal{R}(\mathcal{A}) \subseteq \phi^+_T(\mathcal{R}(\mathcal{B}))$$

We want to show that

$$\mathcal{R}(\mathcal{A}\cdot N) \subseteq \phi^+_T(\mathcal{R}(\mathcal{B}\cdot N)),$$

which is of course trivial thanks to requirement I of Definition 42.

Theorem 39. Requirement III of Definition 42 exists to support this axiom. It says:

$$\mathcal{R}(\mathcal{A}\wedge\mathcal{B})\cdot N \subseteq \mathcal{R}(\mathcal{A}\cdot N) \cup \mathcal{R}(\mathcal{B}\cdot N)$$

The right-hand side, by the semantics for $\wedge$, is equal to $\mathcal{R}((\mathcal{A}\cdot N) \wedge (\mathcal{B}\cdot N))$, completing the proof.

Axiom 40. Since $(\mathcal{A}\wedge\mathcal{B}) \Rightarrow \mathcal{A}$, $(\mathcal{A}\wedge\mathcal{B})\cdot N \Rightarrow \mathcal{A}\cdot N$ (by Axiom 38, with $T = \mathcal{U}$). The same is true for $\mathcal{B}$, proving:

$$(\mathcal{A}\wedge\mathcal{B})\cdot N \Rightarrow (\mathcal{A}\cdot N) \wedge (\mathcal{B}\cdot N) \qquad \square$$

Theorem 41. Theorem 39 and Axiom 40 together show equality. □

Axiom 44. Assume $\mathcal{R}(\mathcal{B}) \subseteq \mathcal{E}(\sigma') \supset \mathcal{R}(\mathcal{A}) \subseteq \mathcal{E}(\sigma')$ for $\sigma' \in \{\sigma, \tau\}$. Further, assume that $\mathcal{R}(\mathcal{B}) \subseteq \mathcal{E}(\sigma\wedge\tau)$. Using the semantics of $\wedge$, we can write $\mathcal{R}(\mathcal{B}) \subseteq \mathcal{E}(\sigma) \cap \mathcal{E}(\tau)$, and hence $\mathcal{R}(\mathcal{B}) \subseteq \mathcal{E}(\sigma)$ and $\mathcal{R}(\mathcal{B}) \subseteq \mathcal{E}(\tau)$. By the first assumption, we can replace $\mathcal{B}$ in both statements with $\mathcal{A}$, use the definition of $\cap$ and the semantics of $\wedge$, and conclude that $\mathcal{R}(\mathcal{A}) \subseteq \mathcal{E}(\sigma\wedge\tau)$, justifying the axiom. □

Axiom 45. Let $T = \{\sigma, \tau\}$ and $T' = \{\sigma\wedge\tau\}$. Assume first that:

$$\phi^w_T(\mathcal{R}(\mathcal{A})(w'_0)) \subseteq \phi^w_T(\mathcal{R}(\mathcal{B})(w'_0))\ \forall\, w'_0 \in W$$

Second, assume we are given w_0 and $\overline{w}'_1$ such that $\overline{w}'_1 \in \phi^w_{T'}(\mathcal{R}(\mathcal{A})(w_0))$. We have the existence of a $w_1 \in \mathcal{R}(\mathcal{A})(w_0)$ with $\overline{w}'_1 = \phi_{T'}(w_1)$.

Let $\overline{w}_1 = \phi_T(w_1)$. By our first assumption, $\overline{w}_1 \in \phi^w_T(\mathcal{R}(\mathcal{B})(w_0))$, so there is a $w'_1 \in \mathcal{R}(\mathcal{B})(w_0)$ with $\overline{w}_1 = \phi_T(w'_1)$. We claim that $\phi_{T'}(w'_1) = \overline{w}'_1$, a claim supported by leaning on the definition of ϕ_T:

$$\begin{aligned}
\sigma \wedge \tau \in \phi_{T'}(w'_1) &\equiv w'_1 \in \mathcal{E}(\sigma \wedge \tau)\\
&\equiv w'_1 \in \mathcal{E}(\sigma) \wedge w'_1 \in \mathcal{E}(\tau)\\
&\equiv \sigma \in \overline{w}_1 \wedge \tau \in \overline{w}_1\\
&\equiv w_1 \in \mathcal{E}(\sigma) \wedge w_1 \in \mathcal{E}(\tau)\\
&\equiv w_1 \in \mathcal{E}(\sigma \wedge \tau)\\
&\equiv \sigma \wedge \tau \in \overline{w}'_1
\end{aligned}$$

Since $\overline{w}'_1$ is either $T = \{\sigma \wedge \tau\}$ or $\varnothing$, we have shown the equality, and that $\overline{w}'_1 \in \phi^w_{T'}(\mathcal{R}(\mathcal{B})(w_0))$. Therefore the model supports $B \overset{\{\sigma\wedge\tau\}}{\Rightarrow} A$. □

Axiom 46. The structure of this proof parallels that of Axiom 45. Let $T = \{\sigma\}$ and $T' = \{\neg\sigma\}$. Assume first that:

$$\phi^w_T(\mathcal{R}(\mathcal{A})(w'_0)) \subseteq \phi^w_T(\mathcal{R}(\mathcal{B})(w'_0)) \;\forall\, w'_0 \in W$$

Second, assume we are given w_0 and $\overline{w}'_1$ such that $\overline{w}'_1 \in \phi^w_{T'}(\mathcal{R}(\mathcal{A})(w_0))$. That implies the existence of a $w_1 \in \mathcal{R}(\mathcal{A})(w_0)$, with $\overline{w}'_1 = \phi_{T'}(w_1)$. By the definition of $\phi_{T'}$ we know $w_1 \in \mathcal{E}(\neg\sigma)$ if and only if $\neg\sigma \in \overline{w}'_1$. Using the semantics of ¬, we can rewrite that expression as

$$w_1 \in \mathcal{E}(\sigma) \;\textbf{iff}\; \neg\sigma \notin \overline{w}'_1$$

Define

$$\overline{w}_1 = \begin{cases} T & \text{if } \overline{w}'_1 = \varnothing \\ \varnothing & \text{otherwise } (\overline{w}'_1 = T') \end{cases}$$

Clearly $\sigma \in \overline{w}_1$ if and only if $\neg\sigma \notin \overline{w}'_1$. Now we can write

$$w_1 \in \mathcal{E}(\sigma) \;\textbf{iff}\; \sigma \in \overline{w}_1$$

This expression satisfies the definition of ϕ_T, so we have $\phi_T(w_1) = \overline{w}_1$. Because $w_1 \in \mathcal{R}(A)(w_0)$, we know $\overline{w}_1 \in \phi^w_T(\mathcal{R}(A)(w_0))$.

Using the first assumption, we have $\overline{w}_1 \in \phi^w_T(\mathcal{R}(B)(w_0))$. Using arguments analogous to those above, we have the existence of a $w'_1 \in \mathcal{R}(B)(w_0)$, and by the definition of ϕ_T, we can show that $\overline{w}'_1$ is in $\phi^w_T(\mathcal{R}(B)(w_0))$ as well. The model supports $\mathcal{B} \overset{\neg\sigma}{\Rightarrow} \mathcal{A}$. □

A.5 Relationships among the restricted relations

In each of the examples below, assume $T = \{s\}$.

$\overset{T}{\Rrightarrow}$ **is not stronger than** $\overset{T}{\Rightarrow}$**.** The subset relation in the projected model $\overline{M}$ of $\overset{T}{\Rightarrow}$ holds with the possible exception of the single world $\overline{w}_T = T$ that represents the equivalence class of worlds in $\mathcal{M}$ in which all statements in T hold. Clearly ϕ_T takes every member of $\cap_{\sigma\in T}\mathcal{E}(\sigma)$ to that representative. The counterexample illustrated in Figure 9 highlights this exception.

$\overset{T}{\Rightarrow}$ **is not stronger than** $\overset{T}{\Rrightarrow}$**.** Although just showed that $\overset{T}{\Rrightarrow}$ is not quite stronger than $\overset{T}{\Rightarrow}$, it certainly seems almost so. Indeed, it is very easy to construct an example that shows that the mighty relation does not follow from the basic speaks-for-regarding relation. See Figure 10.

$\overset{T}{\Rightarrow}$ **implies** $\overset{T}{\rightarrow}$**.** Assume $\mathcal{R}(A) \subseteq \phi^+_T(\mathcal{R}(B))$. We will prove by contradiction that $B \overset{T}{\rightarrow} A$. To establish a contradiction, we assume there is a statement $\sigma \in T$ and a world w_0 where $B\,\mathbf{says}\,\sigma$ but not $A\,\mathbf{says}\,\sigma$. That is, $\mathcal{R}(B)(w_0) \subseteq \mathcal{E}(\sigma)$ but $\mathcal{R}(A)(w_0) \nsubseteq \mathcal{E}(\sigma)$. The latter means that there is a world $w_1 \in \mathcal{R}(A)(w_0)$, but $w_1 \notin \mathcal{E}(\sigma)$.

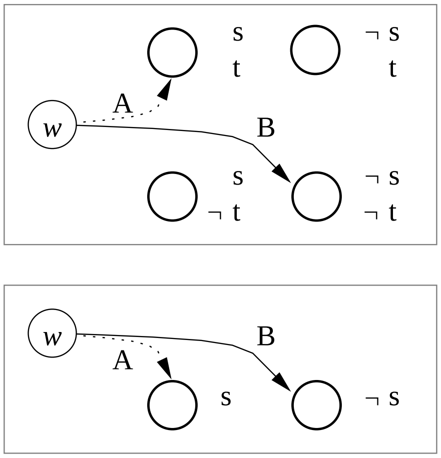
We can push $\langle w_0, w_1\rangle$ through our original assumption to find a w'_1 such that $\langle w_0, w'_1\rangle \in \mathcal{R}(B)$ and $w'_1 \cong_T w_1$. Definition 32 tells us that $w'_1 \notin \mathcal{E}(\sigma)$, which means $\mathcal{R}(B)(w_0) \nsubseteq \mathcal{E}(\sigma)$, which contradicts our second assumption. We may conclude that for all $w_0 \in W$ and $\sigma \in T$, $\mathcal{R}(B)(w_0) \subseteq \mathcal{E}(\sigma)$ implies $\mathcal{R}(A)(w_0) \subseteq \mathcal{E}(\sigma)$. □

$\overset{T}{\Rrightarrow}$ **implies** $\overset{T}{\rightarrow}$**.** We assume

$$\mathcal{R}(A)(w_0) - \bigcap_{\tau\in T}\mathcal{E}(\tau) \subseteq \mathcal{R}(B)(w_0)$$

and that $\mathcal{R}(B)(w_0) \subseteq \mathcal{E}(\sigma)$. From the first assumption, any world $w_1 \in \mathcal{R}(A)(w_0)$ is either in $\mathcal{E}(\sigma)$ (let $\tau = \sigma$) or in $\mathcal{R}(B)(w_0)$. The former case trivially guarantees $w_1 \in \mathcal{E}(\sigma)$, and the latter case does so by the second assumption. We conclude that $\mathcal{R}(A)(w_0) \subseteq \mathcal{E}(\sigma)$. □

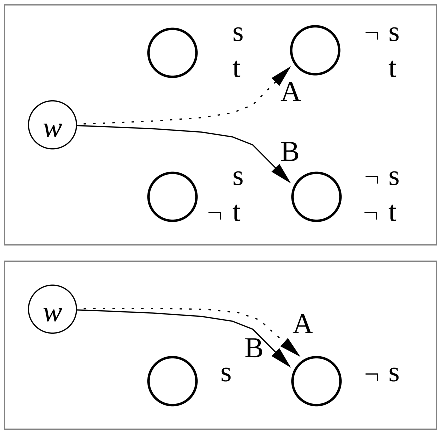
$\overset{T}{\rightarrow}$ **is weaker than** $\overset{T}{\Rightarrow}$ **and** $\overset{T}{\rightarrow}$ **is weaker than** $\overset{T}{\Rrightarrow}$**.** See Figure 11 for counterexamples that illustrate these relationships.



The set $\cap_{s\in T}\mathcal{E}(s)$ is the left pair of worlds (where s is true); the only edge belonging to $\mathcal{R}(A)$ terminates in one of those worlds. Therefore, in this model, $\mathcal{R}(A)(w) - \cap_{s\in T}\mathcal{E}(s) \subseteq \mathcal{R}(B)(w)$, and we conclude that $B \overset{T}{\Rrightarrow} A$.

The mapping ϕ_T that reduces the worlds above to equivalence classes modulo statements in T will make this model $\mathcal{M}'$. $\phi_T^R(\mathcal{R}(A))$ includes an edge to the equivalence class labeled s, but $\phi_T^R(\mathcal{R}(B)(w))$ does not. Therefore, $B \overset{T}{\not\Rightarrow} A$.

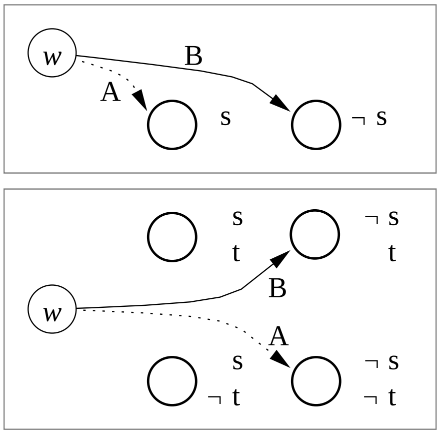
Figure 9: *A counterexample that shows* $B \overset{T}{\Rrightarrow} A$ *does not imply* $B \overset{T}{\Rightarrow} A$.



Here is a model in which from w, A considers possible a world neither in $\mathcal{R}(B)(w)$ nor $\cap_{s\in T}\mathcal{E}(s)$. So $B \overset{T}{\not\Rrightarrow} A$.

Projecting the model onto T, however, shows that $\phi_T^R(\mathcal{R}(A))$ and $\phi_T^R(R(B))$ completely agree on matters related to s; that is, $B \overset{T}{\Rightarrow} A$.

Figure 10: *A counterexample that shows* $B \overset{T}{\Rightarrow} A$ *does not imply* $B \overset{T}{\Rrightarrow} A$.



(a) The statement $(\mathcal{R}(B)(w) \subseteq \mathcal{E}(s)) \supset (\mathcal{R}(A)(w) \subseteq \mathcal{E}(s))$ has a false premise, making it vacuously true in this model. Hence this model satisfies $B \xrightarrow{T} A$. The model is its own projection onto T, however, and it is clear that $B \overset{T}{\not\Rightarrow} A$.

(b) This model satisfies $B \xrightarrow{T} A$ for the same reason as the model in part (a). The single edge terminating at $\mathcal{R}(A)(w)$, however, is in neither $\mathcal{R}(B)(w)$ nor $\cap_{s \in T} \mathcal{E}(s)$, so $B \overset{T}{\not\Rrightarrow} A$.

Figure 11: *Examples that show why the relation $\xrightarrow{T}$ is weaker than $\overset{T}{\Rightarrow}$ and $\overset{T}{\Rrightarrow}$.*