A Modal Deconstruction of Access Control Logics

Deepak Garg[1] and Martín Abadi[2,3]

[1] Carnegie Mellon University
[2] University of California, Santa Cruz
[3] Microsoft Research, Silicon Valley

Abstract. We present a translation from a logic of access control with a "says" operator to the classical modal logic S4. We prove that the translation is sound and complete. We also show that it extends to logics with boolean combinations of principals and with a "speaks for" relation. While a straightforward definition of this relation requires second-order quantifiers, we use our translation for obtaining alternative, quantifier-free presentations. We also derive decidability and complexity results for the logics of access control.

1 Introduction

In computer systems, access control checks restrict the operations that users, machines, and other principals can execute on objects such as files [27]. These checks are governed by access control policies—often by the combination of several policies at different layers and from different entities. In practice, the principals, the objects, the formulations of policies, and their implementations can be quite varied. The resulting gaps, inconsistencies, and obscurity endanger security.

In response to these concerns, specialized logics have been proposed as frameworks for describing, analyzing, and enforcing access control policies (e.g., [2, 3, 6, 10, 19, 20, 29, 30]). A number of research projects have applied these logics for designing or explaining various languages and systems (e.g., [4, 6–10, 13, 14, 16, 18, 26, 29, 35]). On the other hand, there have been only few, limited efforts to study the logics themselves (e.g., [2, 3, 19, 20]). Accordingly, the decidability, expressiveness, and semantics of these logics are largely unexplored.

Our objective in the present paper is to fill this gap. Specifically, we study a class of access control logics via sound and complete translations to the classical modal logic S4.

- Relying on the theory of S4 (e.g., [24, 25]), we obtain Kripke semantics for the logics. In the quantifier-free case, we also establish the decidability of the logics and their PSPACE complexity. The translations also open the possibility of re-using existing decision procedures for S4.
- Translating several logics to S4 enables us to compare their expressiveness. In particular, while a straightforward definition of the "speaks for" relation [26, 28] requires second-order quantifiers, we use our translations for obtaining

alternative, quantifier-free presentations. We prove that these quantifier-free presentations yield the same consequences as the second-order definition.
– The translations also suggest a logic with a boolean structure on principals. Although propositional, this new logic is rich and quite expressive. Previous logics with similar constructs allowed conjunctions and disjunctions of principals (but not negations); the present logic goes beyond them in ways that we consider both elegant and useful.

Access control logics (those studied here and most of those in the literature) include formulas of the form A `says` s, where A is a principal and s is a formula. Intuitively, A `says` s means that A asserts (or supports) s. For example, the administrator admin of a domain might certify that Alice is an authorized user; this assertion may be represented as admin `says` `auth_user`(Alice). In addition, many logics support the use of the "speaks for" relation: $\mathsf{A} \Rightarrow \mathsf{B}$ means that A speaks for B, that is, A `says` s implies B `says` s for every s. For example, when $\mathsf{Key}_{\mathsf{Alice}}$ represents the public key of Alice, one may write $\mathsf{Key}_{\mathsf{Alice}} \Rightarrow \mathsf{Alice}$. When a server S acts on Alice's behalf impersonating her, one may also write $\mathsf{S} \Rightarrow \mathsf{Alice}$. Despite these similarities, logics differ in their axioms. A 2003 survey discusses some of the options [1]. Recently, several works [2, 19, 20, 29] have basically relied upon the rules of lax logic and the computational lambda calculus [11, 17, 33] for the operator `says`. This approach has several benefits, for example validating the "handoff axiom" [2, 26]; a detailed discussion of its features is beyond the scope of this paper. We follow this approach in the logics that we consider.

The first of these logics, called ICL, extends propositional intuitionistic logic with the operator `says` which behaves as a principal-indexed lax modality (Section 2). ICL can be viewed as an indexed version of CL [11], hence its name, and also as the common propositional fragment of CDD [2, Section 8] and other systems [20, 29]. An extension of ICL, called $\mathrm{ICL}^{\Rightarrow}$, allows formulas of the form $\mathsf{A} \Rightarrow \mathsf{B}$ (Section 3). Another extension, called $\mathrm{ICL}^{\mathcal{B}}$, allows compound principals formed with boolean connectives (Section 4). Our translations and the resulting theorems apply to each of these logics. In addition, we show that $\mathsf{A} \Rightarrow \mathsf{B}$ can be encoded using either compound principals or a second-order universal quantifier (Sections 5 and 6). We conclude with a discussion of directions for further work (Section 7). Proofs are available on-line at `www.cs.cmu.edu/~dg/papers/modal-decons-full.pdf`.

Related Work. Our translations are partly based on a translation from intuitionistic logic to S4 that goes back to Gödel [22]. Moreover, ICL can be seen as a rather direct generalization of lax logic. Nevertheless, our translation from ICL (and, as a special case, from lax logic) to S4 appears to be new.

Partly following Curry [15], Fairtlough and Mendler suggested interpreting lax logic in intuitionistic logic by mapping $\bigcirc s$ to $C \vee s$ or to $C \supset s$, where $\bigcirc$ is a lax modality and C is a fixed proposition [17]. These interpretations are sound but not complete. Composing them with a translation from intuitionistic logic to S4, one can map $\bigcirc s$ to $\Box((\Box C) \vee s)$ or to $\Box((\Box C) \supset s)$. A similar translation from lax logic to S4 follows from our definitions, as a special case; however, our translation does not put a $\Box$ on C, and it is sound and complete.

Other interpretations of lax logic have targeted multimodal logics or intuitionistic S4 [5, 11, 17, 34]. Our translations seem simpler; in particular, they target classical S4. Semantically, those interpretations lead to Kripke models with at least two accessibility relations, while we need only one.

Fairtlough and Mendler also deduced the decidability of lax logic from a subformula property [17]. Further, Howe developed a PSPACE decision procedure for lax logic [23]. It seems possible to extend Howe's approach to obtain an alternative proof of decidability for ICL. We do not know whether it would also apply to richer logics such as $\text{ICL}^{\Rightarrow}$ and $\text{ICL}^{\mathcal{B}}$, for which we have not established a subformula property.

Going beyond basic lax logic, not much is known about the theory of logics with compound principals or with a "speaks for" relation (such as $\text{ICL}^{\Rightarrow}$ and $\text{ICL}^{\mathcal{B}}$). Some of the early work on the subject started to explore semantics and decidability results [3]. Although sometimes helpful, the semantics were not sound and complete, and the decidability results applied only to fragments needed for certain access-control decisions. More recent systems like RT and SecPAL (where the "can act as" relation resembles $\Rightarrow$) include decision procedures for useful classes of formulas similar to Horn clauses [10, 31, 32].

2 ICL: A Basic Logic of Access Control

We start with a basic access control logic ICL that includes the operator `says` but not $\Rightarrow$. Although minimal in its constructs, the logic is reasonably expressive. We describe a translation from ICL to classical S4. From this translation we derive a Kripke semantics and a decidability result.

2.1 The Logic

Formulas in ICL may be atomic propositions (p, q, etc.) or constructed from standard connectives $\wedge$ (conjunction), $\vee$ (disjunction), $\supset$ (implication), $\top$ (true), and $\bot$ (false), and the operator `says`.

$$s \ ::= \ p \mid s_1 \wedge s_2 \mid s_1 \vee s_2 \mid s_1 \supset s_2 \mid \top \mid \bot \mid \mathsf{A}\ \texttt{says}\ s$$

The letters A, B, etc., denote principals, which are atomic and distinct from atomic propositions. They may be simple bit-string representations of names; in Section 4, we generalize principals to a richer algebra.

ICL inherits all the inference rules of intuitionistic propositional logic, which we elide here. For each principal A, the formula $\mathsf{A}\ \texttt{says}\ s$ satisfies the following axioms:

$$\begin{array}{ll} \vdash s \supset (\mathsf{A}\ \texttt{says}\ s) & \text{(unit)} \\ \vdash (\mathsf{A}\ \texttt{says}\ (s \supset t)) \supset (\mathsf{A}\ \texttt{says}\ s) \supset (\mathsf{A}\ \texttt{says}\ t) & \text{(cuc)} \\ \vdash (\mathsf{A}\ \texttt{says}\ \mathsf{A}\ \texttt{says}\ s) \supset \mathsf{A}\ \texttt{says}\ s & \text{(idem)} \end{array}$$

These mean that $\mathsf{A}\ \texttt{says}\ \cdot$ is a lax modality [17]. We describe them briefly, referring the reader to the literature on lax logic and computational lambda calculus for more details and applications.

- (unit) states that every true formula s is supported by every principal A. (The converse is not true: principals may make false statements.)
- (cuc) allows us to reason with A's statements. It says that whenever A states $s \supset t$ and s, it also states t. Thus A's statements are closed under logical consequence.
- (idem) collapses applications of $\mathsf{A}\ \mathtt{says}\ \cdot$. In the context of (unit), (idem) implies that $\mathsf{A}\ \mathtt{says}\ \cdot$ is idempotent.

Example 1 We illustrate the use of ICL through a simple example. Consider a file-access scenario with an administrating principal admin, a user Bob, one file file1, and the following policy:

1. If admin says that file1 should be deleted, then this must be the case.
2. admin trusts Bob to decide whether file1 should be deleted.
3. Bob wants to delete file1.

Intuitively, from these facts we should be able to conclude that file1 should be deleted. We describe a logical presentation of this example in ICL. Suppose that the proposition deletefile1 means that file1 should be deleted. The three facts above can be written:

1. (admin says deletefile1) $\supset$ deletefile1
2. admin says ((Bob says deletefile1) $\supset$ deletefile1)
3. Bob says deletefile1

Using (unit) and (cuc), (1)–(3) imply deletefile1.

2.2 Translation from ICL to S4

Next we describe a central technical result of our work: a sound and complete translation from ICL to S4. Before describing the translation, we briefly sketch S4. More details may be found in standard references (e.g., [24]); S4 has been studied thoroughly over the years.

S4. S4 is an extension of classical logic with one modality $\Box$, and the rules:

From $\vdash s$ infer $\vdash \Box s$.
$\vdash \Box (s \supset t) \supset \Box s \supset \Box t$
$\vdash \Box s \supset s$
$\vdash \Box s \supset \Box \Box s$

Translation. Our translation $\ulcorner \cdot \urcorner$ from ICL to S4 is summarized in Figure 1. It is defined by induction on the structure of formulas. For atomic formulas and non-modal connectives, the translation is a slight simplification of Gödel's translation from intuitionistic logic to S4 [22]. (In Gödel's words, the basic idea is to "put a box around everything"; we simplify the translation by putting

$$
\begin{array}{rcl}
\ulcorner p \urcorner & = & \Box\, p \\
\ulcorner s \wedge t \urcorner & = & \ulcorner s \urcorner \wedge \ulcorner t \urcorner \\
\ulcorner s \vee t \urcorner & = & \ulcorner s \urcorner \vee \ulcorner t \urcorner \\
\ulcorner s \supset t \urcorner & = & \Box\,(\ulcorner s \urcorner \supset \ulcorner t \urcorner) \\
\ulcorner \top \urcorner & = & \top \\
\ulcorner \bot \urcorner & = & \bot \\
\ulcorner \mathsf{A}\ \mathtt{says}\ s \urcorner & = & \Box\,(\mathsf{A} \vee \ulcorner s \urcorner)
\end{array}
$$

Fig. 1. Translation from ICL to S4

boxes only around atomic formulas and implications.) The core of our work is the translation of $\mathsf{A}\ \mathtt{says}\ s$.

$$\ulcorner \mathsf{A}\ \mathtt{says}\ s \urcorner \ = \ \Box\,(\mathsf{A} \vee \ulcorner s \urcorner)$$

We interpret the principal A as an atomic formula in S4 and assume that such atomic formulas are distinct from the usual atomic formulas p, q, etc.. Informally, if we read $\Box$ as "in all possible worlds" and the atomic formula A as "principal A is unhappy", then $\Box\,(\mathsf{A} \vee \ulcorner s \urcorner)$ means that $\ulcorner s \urcorner$ holds in all possible worlds in which A is happy.

Alternatively, but equivalently, we could set: $\ulcorner \mathsf{A}\ \mathtt{says}\ s \urcorner \ = \ \Box\,(\mathsf{A} \supset \ulcorner s \urcorner)$. Since the target of the translation is a classical logic, the difference between $\Box\,(\mathsf{A} \vee \ulcorner s \urcorner)$ and $\Box\,(\mathsf{A} \supset \ulcorner s \urcorner)$ is only superficial. We prefer $\Box\,(\mathsf{A} \vee \ulcorner s \urcorner)$ because it leads to a more memorable interpretation of $\Rightarrow$ in Section 3.

This simple translation is correct in the sense that it is both sound and complete:

Theorem 1 (Soundness and Completeness) *For every ICL formula s, $\vdash s$ in ICL if and only if $\vdash \ulcorner s \urcorner$ in S4.*

The proof of this theorem relies heavily on the proof theory of ICL and S4.

2.3 Decidability and Kripke Models for ICL

Decidability is a desirable property in an access control logic: it allows the possibility of completely automated tools for analyzing policies. In the case of ICL, Theorem 1 implies PSPACE decidability since the same complexity bound is known for S4 [25]. This bound is the best we could expect, since PSPACE-hardness holds for plain intuitionistic propositional logic.

Corollary 1 (Decidability) *There is a polynomial space procedure that decides whether a given ICL formula is provable or not.*

Kripke models are attractive for access control logics from several perspectives. First, they provide a semantic grounding of the logics. They are also useful as mathematical objects, for instance for showing that certain formulas are not derivable. We use Theorem 1 and standard models of S4 to derive Kripke models for ICL.

Definition 1 (Kripke Models) A Kripke model for ICL is a tuple $\langle W, \leq, \rho, \theta \rangle$ where

- W is a set, whose elements are called *worlds* (denoted using the letter w and its decorated variants).
- $\leq$ is a binary relation on W called the *accessibility relation*. When $w \leq w'$, we say that w' is accessible from w. We assume that $\leq$ is reflexive and transitive. We often write $\geq$ for $(\leq)^{-1}$.
- ρ is a mapping from atomic formulas of ICL to $\mathcal{P}(W)$ (the power set of W), called the *assignment*. Intuitively, $\rho(p)$ is the set of worlds in which p holds. We assume that ρ is hereditary with respect to $\leq$, that is, if $w \in \rho(p)$, then for all w' such that $w' \geq w$, $w' \in \rho(p)$.
- θ is a mapping from principals of ICL to $\mathcal{P}(W)$, called the *view map*. When $w \in \theta(\mathsf{A})$, we say that w is *invisible* to A, else it is *visible* to A.

Definition 2 (Satisfaction) Given an ICL formula s and a Kripke model $\mathcal{K} = \langle W, \leq, \rho, \theta \rangle$, we define the satisfaction relation at a particular world ($w \models s$) by induction on s.

- $w \models p$ iff $w \in \rho(p)$
- $w \models s \wedge t$ iff $w \models s$ and $w \models t$
- $w \models s \vee t$ iff $w \models s$ or $w \models t$
- $w \models s \supset t$ iff for each $w' \geq w$, $w' \models s$ implies $w' \models t$
- $w \models \top$ for every w
- not$(w \models \bot)$ for every w
- $w \models \mathsf{A}$ says s iff for every $w' \geq w$, either $w' \in \theta(\mathsf{A})$ or $w' \models s$

Thus, this definition implies that a world satisfies A says s iff every reachable world that is visible to A satisfies s. For other constructs, the definition of satisfaction mirrors that in standard Kripke models of intuitionistic logic.

We say that $\mathcal{K} = \langle W, \leq, \rho, \theta \rangle \models s$ if $w \models s$ for every $w \in W$. A formula s is *valid* (written $\models s$) if $\mathcal{K} \models s$ for every Kripke model $\mathcal{K}$. The following result shows that provability in ICL coincides with validity.

Corollary 2 *For every ICL formula s, $\vdash s$ if and only if $\models s$.*

3 ICL$^{\Rightarrow}$: A Logic with A Primitive "Speaks For" Relation

In this section we extend the logic ICL to include a primitive "speaks for" relation. We call the new logic ICL$^{\Rightarrow}$. We also extend the results of Section 2 to ICL$^{\Rightarrow}$.

3.1 The Logic

ICL$^{\Rightarrow}$ extends ICL with formulas of the form $\mathsf{A} \Rightarrow \mathsf{B}$ and with the following axioms for these formulas:

$\vdash \mathsf{A} \Rightarrow \mathsf{A}$	(refl)
$\vdash (\mathsf{A} \Rightarrow \mathsf{B}) \supset (\mathsf{B} \Rightarrow \mathsf{C}) \supset (\mathsf{A} \Rightarrow \mathsf{C})$	(trans)
$\vdash (\mathsf{A} \Rightarrow \mathsf{B}) \supset (\mathsf{A}$ says $s) \supset (\mathsf{B}$ says $s)$	(speaking-for)
$\vdash (\mathsf{B}$ says $(\mathsf{A} \Rightarrow \mathsf{B})) \supset (\mathsf{A} \Rightarrow \mathsf{B})$	(handoff)

– (refl) and (trans) state that $\Rightarrow$ is reflexive and transitive.
– (speaking-for) states that if $\mathsf{A} \Rightarrow \mathsf{B}$ and A says s, then B says s.
– (handoff) states that whenever B says that A speaks for B, then A does indeed speak for B. This axiom allows every principal to decide which principals speak on its behalf [26].

Example 2 We modify Example 1: instead of having Bob says deletefile1 directly, Bob delegates his authority to Alice (fact 3), who wants to delete file1 (fact 4).

1. (admin says deletefile1) $\supset$ deletefile1
2. admin says ((Bob says deletefile1) $\supset$ deletefile1)
3. Bob says Alice $\Rightarrow$ Bob
4. Alice says deletefile1

Using (handoff) and (speaking-for), we can again derive deletefile1.

3.2 Translation from ICL$^{\Rightarrow}$ to S4

We extend to ICL$^{\Rightarrow}$ the translation from ICL to S4 by adding the clause:

$$\ulcorner \mathsf{A} \Rightarrow \mathsf{B} \urcorner \;=\; \Box(\mathsf{A} \supset \mathsf{B})$$

As above, A and B are interpreted as atomic formulas in S4, and these atomic formulas are assumed distinct from the atomic propositions of ICL$^{\Rightarrow}$. We have:

Theorem 2 (Soundness and Completeness) *For every ICL$^{\Rightarrow}$ formula s, $\vdash s$ in ICL$^{\Rightarrow}$ if and only if $\vdash \ulcorner s \urcorner$ in S4.*

3.3 Decidability and Kripke Models for ICL$^{\Rightarrow}$

Much as for ICL, Theorem 2 yields a decidability result:

Corollary 3 (Decidability) *There is a polynomial space procedure that decides whether a given ICL$^{\Rightarrow}$ formula is provable or not.*

It also leads to Kripke models for ICL$^{\Rightarrow}$. These are the same as those for ICL (Definition 1), with the satisfaction relation for $\mathsf{A} \Rightarrow \mathsf{B}$ at world w given by the clause:

$w \models \mathsf{A} \Rightarrow \mathsf{B}$ iff for every $w' \geq w$, $w' \in \theta(\mathsf{A})$ implies $w' \in \theta(\mathsf{B})$.

These models are sound and complete in the sense of Corollary 2.

4 ICL$^{\mathcal{B}}$: A Logic with Boolean Principals

Principals in ICL and ICL$^{\Rightarrow}$ are atomic and cannot be composed in any logically meaningful way. Early on it was observed that the use of compound principals can help in expressing policies [3, 26]. For example, the conjunction of two principals may be employed for representing joint statements, with the property

$$(\mathsf{A} \wedge \mathsf{B})\ \mathtt{says}\ s \equiv (\mathsf{A}\ \mathtt{says}\ s) \wedge (\mathsf{B}\ \mathtt{says}\ s)$$

Disjunctions also arise, though they are more complex. Going further, we describe and study a systematic extension ICL$^{\mathcal{B}}$ of ICL that allows arbitrary Boolean combinations of principals with the connectives $\wedge$, $\vee$, $\supset$, $\top$, and $\bot$. (However, we do not include operators such as "quoting" and "for".) We extend the results of Section 2 to ICL$^{\mathcal{B}}$.

4.1 The Logic

The formulas of ICL$^{\mathcal{B}}$ are the same as those of ICL, except that principals may contain Boolean connectives. We use the letters a, b, ... for denoting atomic principals (distinct from atomic propositions), and A, B, ... for denoting arbitrary principals.

$$\mathsf{A}, \mathsf{B} \ ::= \ a \mid \mathsf{A} \wedge \mathsf{B} \mid \mathsf{A} \vee \mathsf{B} \mid \mathsf{A} \supset \mathsf{B} \mid \top \mid \bot$$

We write $\neg\mathsf{A}$ for $(\mathsf{A} \supset \bot)$. We equip the set of principals with a notion of equality by letting $\mathsf{A} \equiv \mathsf{B}$ if A and B are provably equivalent when viewed as formulas in classical logic. With these definitions, the set of principals becomes a Boolean algebra.

ICL$^{\mathcal{B}}$ inherits all the inference rules of ICL, and also includes the following additional rules:

$\vdash (\bot\ \mathtt{says}\ s) \supset s$	(trust)
If $\mathsf{A} \equiv \top$ then $\vdash \mathsf{A}\ \mathtt{says}\ \bot$.	(untrust)
$\vdash ((\mathsf{A} \supset \mathsf{B})\ \mathtt{says}\ s) \supset (\mathsf{A}\ \mathtt{says}\ s) \supset (\mathsf{B}\ \mathtt{says}\ s)$	(cuc')

- (trust) states that $\bot$ is a truth teller.
- (untrust) states that any principal equivalent to $\top$ says false; it can be seen as a variant of the necessitation rule of modal logics.
- Similarly, (cuc') is the analogue of (cuc) for principals. It states that $\mathsf{A}\ \mathtt{says}\ s$ and $(\mathsf{A} \supset \mathsf{B})\ \mathtt{says}\ s$ imply $\mathsf{B}\ \mathtt{says}\ s$.

We define ICL$^{\mathcal{B}}$ as an extension of ICL, rather than ICL$^{\Rightarrow}$, because we do not need built-in formulas of the form $\mathsf{A} \Rightarrow \mathsf{B}$. The "speaks for" relation is definable in ICL$^{\mathcal{B}}$. As we show in Section 5, $\mathsf{A} \Rightarrow \mathsf{B}$ can be seen as an abbreviation for $(\mathsf{A} \supset \mathsf{B})\ \mathtt{says}\ \bot$.

We can explain the intuitive meaning of $\mathsf{A}\ \mathtt{says}\ s$ when principal A is compound, as follows:

- $(\mathsf{A} \wedge \mathsf{B})\ \mathtt{says}\ s$ is the same as $(\mathsf{A}\ \mathtt{says}\ s) \wedge (\mathsf{B}\ \mathtt{says}\ s)$.

- $(\mathsf{A} \vee \mathsf{B})$ says s means that, by combining the statements of A and B, we can conclude s. In particular, if A says $(s \supset t)$ and B says s then $(\mathsf{A} \vee \mathsf{B})$ says t. Disjunctions can be used in modeling groups in access control.
- $(\mathsf{A} \supset \mathsf{B})$ says s means that A speaks for B on s and on its consequences. We can show that if $(\mathsf{A} \supset \mathsf{B})$ says s and $s \supset s'$, then $(\mathsf{A}$ says $s') \supset (\mathsf{B}$ says $s')$. In the special case where s is $\bot$, we obtain the usual $\Rightarrow$ relation.
- $\top$ says s is provable for every formula s (including $\bot$). In access control terms, $\top$ may be seen as a completely untrustworthy principal.
- $\bot$ says s implies that s is true. Thus, $\bot$ is a completely trustworthy principal.

Example 3 The following policy is analogous to that of Example 1:

1. $(\mathsf{admin} \supset \bot)$ says deletefile1
2. admin says $(\mathsf{Bob} \supset \mathsf{admin})$ says deletefile1
3. Bob says deletefile1

The first statement means that admin is trusted on deletefile1 and its consequences. The second statement means that admin further delegates this authority to Bob.

From (3) and (unit) it follows that admin says Bob says deletefile1. From (2), (cuc), and (cuc') we get (admin says Bob says deletefile1) $\supset$ (admin says admin says deletefile1). Hence we have admin says admin says deletefile1. Using (idem), we obtain admin says deletefile1. From (1) and (cuc'), we obtain (admin says deletefile1) $\supset \bot$ says deletefile1, and hence $\bot$ says deletefile1. Finally, using (trust), we conclude deletefile1.

4.2 Translation from ICL$^{\mathcal{B}}$ to S4

The translation from ICL to S4 works virtually unchanged for ICL$^{\mathcal{B}}$. In the clause $\ulcorner \mathsf{A} \text{ says } s \urcorner = \Box(\mathsf{A} \vee \ulcorner s \urcorner)$, we interpret A as a formula in S4 in the most obvious way: each Boolean connective in A is mapped to the corresponding connective in S4, and each atomic principal in A is interpreted as an atomic formula in S4 (without any added boxes). For instance, the translation of

$$(\mathsf{Bob} \supset \mathsf{admin}) \text{ says deletefile1}$$

is

$$\Box((\mathsf{Bob} \supset \mathsf{admin}) \vee \Box \mathsf{deletefile1})$$

Again, we have soundness and completeness results:

Theorem 3 (Soundness and Completeness) *For every ICL$^{\mathcal{B}}$ formula s, $\vdash s$ in ICL$^{\mathcal{B}}$ if and only if $\vdash \ulcorner s \urcorner$ in S4.*

4.3 Decidability and Kripke Models for ICL$^{\mathcal{B}}$

Once more we obtain a decidability result:

Corollary 4 (Decidability) *There is a polynomial space procedure that decides whether a given ICL$^{\mathcal{B}}$ formula is provable or not.*

Furthermore, Kripke models for ICL$^{\mathcal{B}}$ may be obtained by generalizing those for ICL. The view map θ is defined only for atomic principals a. It is lifted to the function $\hat{\theta}$ that maps all principals to $\mathcal{P}(W)$ as follows:

$$\begin{array}{rcc} \hat{\theta}(a) & = & \theta(a) \\ \hat{\theta}(\mathsf{A} \wedge \mathsf{B}) & = & \hat{\theta}(\mathsf{A}) \cap \hat{\theta}(\mathsf{B}) \\ \hat{\theta}(\mathsf{A} \vee \mathsf{B}) & = & \hat{\theta}(\mathsf{A}) \cup \hat{\theta}(\mathsf{B}) \\ \hat{\theta}(\mathsf{A} \supset \mathsf{B}) & = & (W - \hat{\theta}(\mathsf{A})) \cup \hat{\theta}(\mathsf{B}) \\ \hat{\theta}(\top) & = & W \\ \hat{\theta}(\bot) & = & \emptyset \end{array}$$

The definition of satisfaction ($w \models s$) is modified to use $\hat{\theta}$ instead of θ:

$w \models \mathsf{A}\ \mathtt{says}\ s$ iff for all $w' \geq w$, either $w' \in \hat{\theta}(\mathsf{A})$ or $w' \models s$.

Again, these Kripke models are sound and complete in the sense of Corollary 2.

Thus, while the analysis of the translations requires special (and often difficult) arguments for each logic, the way in which decidability and semantics follow from translations is almost identical across logics. In the remainder of the paper, we turn to more unexpected consequences of the translations.

5 From ICL$^{\Rightarrow}$ to ICL$^{\mathcal{B}}$: Expressing "Speaks For" via Boolean Principals

We prove that $\mathsf{A} \Rightarrow \mathsf{B}$ can be encoded as $(\mathsf{A} \supset \mathsf{B})\ \mathtt{says}\ \bot$. More precisely, we analyze the following translation $(\overline{\cdot})$ from ICL$^{\Rightarrow}$ to ICL$^{\mathcal{B}}$. It maps every connective except $\Rightarrow$ to itself.

$$\begin{array}{rcc} \overline{p} & = & p \\ \overline{s \wedge t} & = & \overline{s} \wedge \overline{t} \\ \overline{s \vee t} & = & \overline{s} \vee \overline{t} \\ \overline{s \supset t} & = & \overline{s} \supset \overline{t} \\ \overline{\top} & = & \top \\ \overline{\bot} & = & \bot \\ \overline{\mathsf{A}\ \mathtt{says}\ s} & = & \mathsf{A}\ \mathtt{says}\ \overline{s} \\ \overline{\mathsf{A} \Rightarrow \mathsf{B}} & = & (\mathsf{A} \supset \mathsf{B})\ \mathtt{says}\ \bot \end{array}$$

(Alternatively, we could translate an extension of ICL$^{\mathcal{B}}$ with $\Rightarrow$ to ICL$^{\mathcal{B}}$.) The encoding of $\Rightarrow$ is correct, in the following sense:

Theorem 4 *For every ICL$^{\Rightarrow}$ formula s, $\vdash s$ in ICL$^{\Rightarrow}$ if and only if $\vdash \overline{s}$ in ICL$^{\mathcal{B}}$.*

This theorem is easy to establish using the translations from ICL$^{\Rightarrow}$ and ICL$^{\mathcal{B}}$ to S4. First we show that for every formula s in ICL$^{\Rightarrow}$, $\ulcorner s \urcorner$ and $\ulcorner \overline{s} \urcorner$ are provably equivalent in S4. This argument is by a structural induction on s. The only interesting case is for s of the form $\mathsf{A} \Rightarrow \mathsf{B}$, where we observe that $\ulcorner \mathsf{A} \Rightarrow \mathsf{B} \urcorner = \Box(\mathsf{A} \supset \mathsf{B}) \equiv \Box((\mathsf{A} \supset \mathsf{B}) \vee \bot) = \ulcorner \overline{\mathsf{A}} \Rightarrow \overline{\mathsf{B}} \urcorner$. It then follows from Theorems 2 and 3 that $\vdash s$ iff $\vdash \ulcorner s \urcorner$ iff $\vdash \ulcorner \overline{s} \urcorner$ iff $\vdash \overline{s}$.

6 On Second-Order Quantification

In this section we consider a logic with second-order quantification. In this logic, $\mathsf{A} \Rightarrow \mathsf{B}$ has a well-known, compelling definition, as an abbreviation for

$$\forall X.\ \mathsf{A}\ \mathtt{says}\ X \supset \mathsf{B}\ \mathtt{says}\ X$$

Our main technical goal is to relate this definition to the quantifier-free axiomatizations of Sections 3–5. We prove that those axiomatizations are sound and complete with respect to the second-order definition. Thus, the full power and complexity of second-order quantification is not required for reasoning about $\Rightarrow$. A decidable fragment of the second-order logic suffices.

(This result was far from obvious to us: a priori, it seemed entirely possible that the axiomatizations were missing some subtle consequence of the second-order definition. Its proof was also surprising, as it includes a non-constructive detour through Kripke models, thus leveraging the work of Sections 3–5.)

6.1 The Logic

The second-order logic is the straightforward extension of ICL with universal quantification over propositions, with the rules of System F [12, 21].

This logic is not entirely new. It has previously been defined [2, Section 8] and used [18] under the name CDD (with only minor syntactic differences). Here we call it ICL$^{\forall}$ for the sake of uniformity.

The addition of second-order quantification provides great expressiveness, as illustrated by the definition of $\Rightarrow$ given above. On the other hand, it immediately leads to undecidability as well as to other difficulties. Nevertheless, this logic is an obvious and elegant extension of ICL.

6.2 Main Results

Though we do not discuss the theory of ICL$^{\forall}$ in detail, we have had to develop some of it in the course of our study of $\Rightarrow$. In this section we present only our main result on $\Rightarrow$ and mention other developments to the extent that they are relevant to this result.

There is an obvious embedding of $\mathrm{ICL}^{\Rightarrow}$ into $\mathrm{ICL}^{\forall}$:

$$\begin{array}{rcl} [\![p]\!] & = & p \\ [\![s \wedge t]\!] & = & [\![s]\!] \wedge [\![t]\!] \\ [\![s \vee t]\!] & = & [\![s]\!] \vee [\![t]\!] \\ [\![s \supset t]\!] & = & [\![s]\!] \supset [\![t]\!] \\ [\![\top]\!] & = & \top \\ [\![\bot]\!] & = & \bot \\ [\![\mathsf{A}\ \mathtt{says}\ s]\!] & = & \mathsf{A}\ \mathtt{says}\ [\![s]\!] \\ [\![\mathsf{A} \Rightarrow \mathsf{B}]\!] & = & \forall X.\ \mathsf{A}\ \mathtt{says}\ X \supset \mathsf{B}\ \mathtt{says}\ X \end{array}$$

This embedding is correct, in the following sense:

Theorem 5 *For every $\mathrm{ICL}^{\Rightarrow}$ formula s, $\vdash s$ in $\mathrm{ICL}^{\Rightarrow}$ if and only if $\vdash [\![s]\!]$ in $\mathrm{ICL}^{\forall}$.*

Soundness (the implication from left to right) is easy to establish. It suffices to show that each axiom of $\mathrm{ICL}^{\Rightarrow}$ can be simulated in $\mathrm{ICL}^{\forall}$ after translation.

Completeness (the implication from right to left) is much harder. Complications arise because a proof of $[\![s]\!]$ may contain formulas which are not in the image of $[\![\cdot]\!]$. Even if we wish to restrict attention to a fragment in which the universal quantifier is restricted to formulas of the form $\forall X.\ \mathsf{A}\ \mathtt{says}\ X \supset \mathsf{B}\ \mathtt{says}\ X$, the proofs of theorems in this fragment may mention formulas that contain universal quantifiers in other positions. Although it is conceivable that a constructive proof-theoretic argument would be viable, this difficulty leads us to a non-constructive argument through acyclic Kripke models.

Our approach seems to be new, so we discuss it in some detail. It is as follows.

- First we define a translation from $\mathrm{ICL}^{\forall}$ to second-order S4 (called $\mathrm{S4}^{\forall}$), that is, classical S4 with a second-order universal quantifier. Let us call this translation $\ulcorner\cdot\urcorner$. This translation essentially mimics the translation of ICL to S4, and in addition maps $\forall X.\ s$ to $\Box\forall X.\ \ulcorner s\urcorner$.
  We show that this translation is sound, in the sense that $\vdash s$ in $\mathrm{ICL}^{\forall}$ implies $\vdash \ulcorner s\urcorner$ in $\mathrm{S4}^{\forall}$. It follows immediately that $\vdash [\![s]\!]$ in $\mathrm{ICL}^{\forall}$ implies $\vdash \ulcorner [\![s]\!]\urcorner$ in $\mathrm{S4}^{\forall}$.
  (We do not need to be concerned about the completeness of this translation for our purposes.)
- Next we may try to show that for every $\mathrm{ICL}^{\Rightarrow}$ formula s, if $\vdash \ulcorner [\![s]\!]\urcorner$ in $\mathrm{S4}^{\forall}$ then $\vdash \ulcorner s\urcorner$ in S4. If this were true, Theorem 2 would yield that $\vdash [\![s]\!]$ in $\mathrm{ICL}^{\forall}$ implies $\vdash s$ in $\mathrm{ICL}^{\Rightarrow}$ (because $\vdash [\![s]\!]$ in $\mathrm{ICL}^{\forall}$ implies $\vdash \ulcorner [\![s]\!]\urcorner$ in $\mathrm{S4}^{\forall}$, as noted above).
  Thus, it would suffice to establish that $\vdash \ulcorner [\![s]\!]\urcorner$ in $\mathrm{S4}^{\forall}$ implies $\vdash \ulcorner s\urcorner$ in S4. We try to prove this by induction on s. Unfortunately, the proof does not go through. The argument fails for a formula of the form $\mathsf{A} \Rightarrow \mathsf{B}$, since

$$\ulcorner [\![\mathsf{A} \Rightarrow \mathsf{B}]\!] \urcorner = \Box\forall X.\ \Box(\Box(\mathsf{A} \vee \Box X) \supset \Box(\mathsf{B} \vee \Box X))$$

  and

$$\ulcorner \mathsf{A} \Rightarrow \mathsf{B} \urcorner = \Box(\mathsf{A} \supset \mathsf{B})$$

In $S4^{\forall}$, the latter implies the former, but the former does not imply the latter.

- Two observations allow the proof to go through:
  1. On all acyclic models, $\ulcorner[\![\mathsf{A} \Rightarrow \mathsf{B}]\!]\urcorner$ implies $\ulcorner \mathsf{A} \Rightarrow \mathsf{B}\urcorner$.
     Therefore, we can establish that all acyclic models satisfy $\ulcorner[\![s]\!]\urcorner$ if and only if all acyclic models satisfy $\ulcorner s\urcorner$.
  2. Quantifier-free S4 is sound and complete with respect to acyclic models. (A model can be "unrolled", and the resulting acyclic model satisfies the same quantifier-free formulas as the original model.)
- Using these observations we can complete our proof as follows.
  - Suppose that $\vdash [\![s]\!]$ in $\mathrm{ICL}^{\forall}$.
  - By the soundness of the translation from $\mathrm{ICL}^{\forall}$ to $S4^{\forall}$, we obtain $\vdash \ulcorner[\![s]\!]\urcorner$ in $S4^{\forall}$.
  - Therefore every acyclic model of $S4^{\forall}$ satisfies $\ulcorner[\![s]\!]\urcorner$.
  - By (1), every acyclic model of $S4^{\forall}$ satisfies $\ulcorner s\urcorner$.
  - Since, for S4 formulas (without quantifiers), the models of $S4^{\forall}$ are the same as the models of S4, every acyclic model of S4 satisfies $\ulcorner s\urcorner$.
  - By (2), every model of S4 satisfies $\ulcorner s\urcorner$.
  - By the completeness of S4 for its models, it follows that $\vdash \ulcorner s\urcorner$ in S4.
  - By Theorem 2, we conclude that $\vdash s$ in $\mathrm{ICL}^{\Rightarrow}$.

7 Conclusion

Starting with a basic logic with a `says` operator, this paper describes simple translations of three logics of access control to S4. The translations lead to decidability results and semantics, and also to comparison of the logics. In particular, the translations enable us to study definitions and axiomatization of the "speaks for" relation.

Going further, one may attempt to carry out a similar programme for some of the diverse logics that appear in the literature. At present, there is no metric to compare these logics against each other, nor a method for integrating more than one logic into a single system. Translation to a standard logic such as S4 seems a promising approach for addressing both of these issues. Of course, first-order and second-order constructs may sometimes be necessary, and more substantial deviations from S4 may arise too—for instance, towards S5, or by the addition of special-purpose operators. Understanding those deviations may be instructive.

Going further, too, our results may be of practical value. They may serve as the basis for theorem provers for logics of access control, with the help of existing algorithms and provers for S4. More speculatively, finite models (of the kind that we obtain from our semantics) may also play a role in a new variant of proof-carrying authentication [6]. A model can serve as evidence that a particular formula is not valid, thus enabling the use of such negative information as an input to authorization decisions. These applications of our results are intriguing; they still require considerable design and experimentation.

Acknowledgments This work was mostly done at Microsoft Research Silicon Valley. The first author was also funded by NSF Grant CNS-0716469. We are grateful to Valeria de Paiva and to Frank Pfenning for discussions on related work.

References

1. Martín Abadi. Logic in access control. In *Proceedings of the 18th Annual Symposium on Logic in Computer Science (LICS'03)*, pages 228–233, June 2003.
2. Martín Abadi. Access control in a core calculus of dependency. *Electronic Notes in Theoretical Computer Science*, 172:5–31, April 2007. *Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin.*
3. Martín Abadi, Michael Burrows, Butler Lampson, and Gordon Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, October 1993.
4. Martín Abadi and Ted Wobber. A logical account of NGSCB. In *Formal Techniques for Networked and Distributed Systems – FORTE 2004*, pages 1–12. Springer-Verlag, 2004.
5. Natasha Alechina, Michael Mendler, Valeria de Paiva, and Eike Ritter. Categorical and kripke semantics for constructive S4 modal logic. In *Computer Science Logic: 15th International Workshop*, pages 292–307. Springer-Verlag, 2001.
6. Andrew W. Appel and Edward W. Felten. Proof-carrying authentication. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 52–62, November 1999.
7. L. Bauer, S. Garriss, J. M. McCune, M. K. Reiter, J. Rouse, and P. Rutenbar. Device-enabled authorization in the Grey system. In *Proceedings of the 8th Information Security Conference (ISC '05)*, pages 431–445, 2005.
8. Lujo Bauer. *Access Control for the Web via Proof-Carrying Authorization.* PhD thesis, Princeton University, November 2003.
9. Lujo Bauer, Scott Garriss, and Michael K. Reiter. Distributed proving in access-control systems. In *Proceedings of the 2005 Symposium on Security and Privacy*, pages 81–95, May 2005.
10. Moritz Y. Becker, Cédric Fournet, and Andrew D. Gordon. Design and semantics of a decentralized authorization language. In *20th IEEE Computer Security Foundations Symposium*, pages 3–15, 2007.
11. P.N. Benton, G.M. Bierman, and V.C.V. de Paiva. Computational types from a logical perspective. *Journal of Functional Programming*, 8(2):177–193, 1998.
12. Luca Cardelli. Type systems. In Allen B. Tucker, editor, *The Computer Science and Engineering Handbook*, chapter 103, pages 2208–2236. CRC Press, Boca Raton, FL, 1997.
13. J. G. Cederquist, R. Corin, M. A. C. Dekker, S. Etalle, J. I. den Hartog, and G. Lenzini. Audit-based compliance control. *Int. J. Inf. Secur.*, 6(2):133–151, 2007.
14. Andrew Cirillo, Radha Jagadeesan, Corin Pitcher, and James Riely. Do as I SaY! programmatic access control with explicit identities. In *20th IEEE Computer Security Foundations Symposium*, pages 16–30, July 2007.
15. Haskell B. Curry. The elimination theorem when modality is present. *Journal of Symbolic Logic*, 17(4):249–265, 1952.

16. John DeTreville. Binder, a logic-based security language. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 105–113, May 2002.
17. M. Fairtlough and M.V. Mendler. Propositional lax logic. *Information and Computation*, 137(1):1–33, August 1997.
18. Cédric Fournet, Andrew D. Gordon, and Sergio Maffeis. A type discipline for authorization in distributed systems. In *20th IEEE Computer Security Foundations Symposium*, pages 31–45, 2007.
19. Deepak Garg, Lujo Bauer, Kevin D. Bowers, Frank Pfenning, and Michael K. Reiter. A linear logic of authorization and knowledge. In *Proceedings of the 11th European Symposium on Computer Security (ESORICS)*, pages 297–312, 2006.
20. Deepak Garg and Frank Pfenning. Non-interference in constructive authorization logic. In *Proceedings of the 19th IEEE Computer Security Foundations Workshop (CSFW 19)*, pages 283–296, 2006.
21. Jean-Yves Girard. *Interprétation Fonctionnelle et Elimination des Coupures de l'Arithmétique d'Ordre Supérieur*. Thèse de doctorat d'état, Université Paris VII, June 1972.
22. Kurt Gödel. Eine interpretation des intuitionistischen aussagenkalkuls. *Ergebnisse eines mathematischen Kolloquiums*, 8:39–40, 1933.
23. Jacob M. Howe. Proof search in lax logic. *Mathematical Structures in Computer Science*, 11(4):573–588, 2001.
24. G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. Methuen Inc., New York, 1968.
25. Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, September 1977.
26. 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, 1992.
27. Butler W. Lampson. Protection. In *Proceedings of the 5th Princeton Conference on Information Sciences and Systems*, pages 437–443, 1971.
28. Butler W. Lampson. Computer security in the real world. *IEEE Computer*, 37(6):37–46, June 2004.
29. Christopher Lesniewski-Laas, Bryan Ford, Jacob Strauss, M. Frans Kaashoek, and Robert Morris. Alpaca: extensible authorization for distributed services. In *14th ACM Conference on Computer and Communications Security*, pages 432–444, 2007.
30. Ninghui Li, Benjamin N. Grosof, and Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security*, 6(1):128–171, February 2003.
31. Ninghui Li and John C. Mitchell. Datalog with constraints: A foundation for trust management languages. In *Proceedings of the Fifth International Symposium on Practical Aspects of Declarative Languages*, pages 58–73, 2003.
32. Ninghui Li, John C. Mitchell, and William H. Winsborough. Beyond proof-of-compliance: security analysis in trust management. *J. ACM*, 52(3):474–514, 2005.
33. Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.
34. Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11:511–540, 2001.
35. 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.