New Modalities for Access Control Logics: Permission, Control and Ratification

Valerio Genovese[1] and Deepak Garg[2]

[1] University of Luxembourg and University of Torino
[2] Carnegie Mellon University

Abstract. We present a new modal access control logic ACL$^+$ to specify, reason about and enforce access control policies. The logic includes new modalities for permission, control, and ratification to overcome some limits of current access control logics. We present a Hilbert-style proof system for ACL$^+$ and a sound and complete Kripke semantics for it. We exploit Kripke semantics to define Seq-ACL$^+$: a sound, complete, cut-free and terminating calculus for ACL$^+$, proving that ACL$^+$ is decidable. We point at a Prolog implementation of Seq-ACL$^+$ and discuss possible extensions of ACL$^+$ with axioms for subordination between principals.

Keywords: Access Control, Delegation, Modal Logic

1 Introduction

Logic plays a prominent role in the specification, reasoning and enforcement of access control policies in distributed systems. In the last two decades, several logic-based formal systems for access control have been proposed (see [1,3] and [7] for surveys), each with its own primitives, semantics and, in some cases, specific application domains. The great variety (and complexity) of such systems makes it difficult to integrate, compare and objectively evaluate them. As is evident from recent research in access control [2,8,11,14,17], modal logic is a powerful framework to study expressiveness, decidability, complexity and semantics of access control logics. Although modal logic has proved useful for theoretical study of access control, it is not widely used in practice to enforce authorization policies (some notable exceptions are [5,6,13,21,22]).

The main reason for this gap is that although several epistemic modalities (e.g., says [18], said and knows [17]) have been studied in the context of access control, key access control concepts like permission, control or trust are not first-class citizens of modal access control languages and must be defined using epistemic modalities. This creates implicit relationships between the concepts and possibly leads to security risks (see [2] for some examples).

In this paper, we take a step towards addressing this shortcoming by proposing a constructive modal logic, ACL$^+$, which extends a standard access control logic with new connectives for permission, control and trust on principals' statements, and admits a decidable calculus. We start by presenting a brief outline of the methodology of access control through logics in Section 2, and a specific connective says [18] that is central to almost all access control logics. In Section 3 we point at three shortcomings of says-based access control logics that, in our opinion, limit their applicability in practical scenarios, thus motivating the need for the new modalities. In Section 4 we present the new modalities, their axioms and inference rules in a Hilbert-style calculus for ACL$^+$. Moreover, we show through examples how ACL$^+$ avoids the shortcomings reported in Section 3.

Section 5 presents sound and complete Kripke semantics for ACL$^+$. Kripke semantics, although useful for establishing several metatheorems of access control logics, are not operational and cannot be used directly in algorithms to reason about authorization problems. Accordingly, in Section 6 we present Seq-ACL$^+$, a sound, complete, cut-free and terminating calculus for ACL$^+$. In Section 7 we present extensions of ACL$^+$ with axioms that force subordination between principals. Section 8 discusses briefly some related work and Section 9 concludes the paper. A full version of this paper with proofs of theorems and an implementation of the decision procedure for ACL$^+$ are available from the authors' webpages.

2 Distributed Access Control Model

We consider a decentralized model of access control, where policy information is *distributed* among several principals. Principals support policy statements and credentials by writing them in certificates signed with their respective private keys. Since policy statements and credentials may be complex, and may assert facts conditional upon statements of other principals, *logic* is a natural choice to model policies. If principal A supports policy (or credential) φ, this is represented in the logic as the formula $A \textsf{ says } \varphi$ [18]. Technically, $A \textsf{ says } \bullet$ is a family of principal-indexed modalities that has been included in several access control logics, albeit with slightly varying semantic interpretations.

An access is authorized (justified) if and only if it is *entailed* by available policy statements and credentials. The question of authorizing an access φ for principal A from a policy Γ can be cast formally as follows: Is it the case that Γ and $A \textsf{ says } \varphi$ entail φ? Or, in symbolic notation, is there a *proof* that $\Gamma, A \textsf{ says } \varphi \vdash \varphi$?

Example 1. Consider the following policy:

1. If the $Admin$ says that $file1$ can be read, then this must be the case[1].
2. $Admin$ trusts Bob to decide whether $file1$ can be read.

In a propositional logic $\mathcal{L}$ enriched with the says modality and equipped with an entailment relation $\vdash_{\mathcal{L}}$, we can express the above policy as follows [2]:

1. $(Admin \textsf{ says } read\_file1) \rightarrow read\_file1$
2. $Admin \textsf{ says } ((Bob \textsf{ says } read\_file1) \rightarrow read\_file1)$

Further, Bob asking to read $file1$ can be represented as $Bob \textsf{ says } read\_file1$. The reference monitor may authorize Bob on this request if and only if

$$(1), (2), Bob \textsf{ says } read\_file1 \vdash_{\mathcal{L}} read\_file1$$

[1] In other words, $Admin$ has direct permission to read $file1$.

In most access control logics, the above entailment has a proof, so Bob will be able to read $file1$. We re-emphasize that the notion of authorization w.r.t. to a submitted request corresponds to the formal notion of *derivability* of the requested access from the available policy.

3 Limits of Access Control Logics: Permissions, Control and Information Flow

In this Section we point out three issues that, in our opinion, create a gap between existing work on logic-based approaches to access control as outlined above, and their deployment in practice. We call the first issue the problem of *implicit permissions*: If an action φ is entailed by a policy Γ, then *any* principal is authorized to perform it. The second issue concerns a logical separation of permission to perform an action from the ability to *control* the action, which also includes the permission to delegate the control further. The third issue is concerned with a fine-grained distinction between the *flow of information* (policy statements) from one principal to another, and its *acceptance* by the receiving principal or, in other words, the issue of separating (in the logic) hearsay from trust in the hearsay. We explain these issues one by one, and then present our proposal to address the issues by introducing new modalities into the logic.

Issue 1: Implicit permissions

The (standard) definition of permission through entailment presented in Section 2 says that a principal A can perform action φ if from the prevailing policy Γ and A says φ, φ can be established. However, this creates a problem in practice: Once enough credentials exist to authorize an access for some principal, any principal is permitted the same access by the standard definition. For instance, in our earlier example, after Bob has created the credential Bob says $read\_file1$, any principal A will be authorized to read $file1$. This is because the existence of a proof of Γ, Bob says $read\_file1 \vdash read\_file1$ implies, by the law of weakening in the logic, that Γ, Bob says $read\_file1, A$ says $read\_file1 \vdash read\_file1$ is also provable for any principal A.

The problem here is that the formula asserting the authorization — $read\_file1$ — does not include the identity of the principal who is authorized access. We propose to resolve this problem by introducing an explicit, principal-indexed modality for permissions, which we write $\mathbf{P}_A\varphi$ (Section 4). With this modality, principal A is authorized to perform action φ iff $\Gamma \vdash \mathbf{P}_A\varphi$. By explicitly listing the principal authorized in the conclusion, we eliminate the problem of implicit permissions.

An alternate, related solution, not considered here, but often used in first-order logics for access control, is to treat the permission (e.g., $read\_file1$) as a relation over principals. Instead of writing $read\_file1$ we could write $read\_file1(A)$ to mean that principal A is authorized to read $file1$. However, since we are interested in proving decidability for the logic, we avoid first-order logic.

Issue 2: Control or Delegatable Permissions

Often in access control, it is desirable to give an individual a permission and also the power to further delegate the permission. To this end we propose a new modality $\mathbf{C}_A\varphi$, read "A controls φ". The key axioms governing $\mathbf{C}_A\varphi$ are:

$$\vdash \mathbf{C}_A\varphi \rightarrow \mathbf{P}_A\varphi \qquad \textit{(C2P)}$$
$$\vdash (\mathbf{C}_A\varphi \wedge (A \textsf{ says } \mathbf{C}_B\varphi)) \rightarrow \mathbf{C}_B\varphi \qquad \textit{(del-C)}$$

The first axiom means that if principal A controls φ, then it is also permitted φ. This axiom relates control to permission and makes $\mathbf{C}_A\varphi$ strictly stronger than $\mathbf{P}_A\varphi$. The second axiom allows principal A, who controls φ, to delegate this control to a principal B simply by asserting this fact. This ability to delegate further distinguishes $\mathbf{C}_A\varphi$ from $\mathbf{P}_A\varphi$.

It is desirable that $(\mathbf{C}_A\varphi_1 \wedge \mathbf{C}_A\varphi_2) \rightarrow \mathbf{C}_A(\varphi_1 \wedge \varphi_2)$. For instance, if A has control over the deletion of files 1 and 2 individually, it should also have control over the deletion of the two files together, thus allowing it to delegate control over deletion of both files at once. A similar property for permissions may be harmful. For instance, if file 2 is the backup of file 1, we may not want to permit their simultaneous deletion ($\mathbf{P}_A(\varphi_1 \wedge \varphi_2)$), even if we allow their deletion individually ($\mathbf{P}_A\varphi_1 \wedge \mathbf{P}_A\varphi_2$). Formally, this difference is manifest in different logical treatments of the two modalities: while $\mathbf{C}_A$ is a normal *necessitation* modality, $\mathbf{P}_A$ is a *possibility* modality (see Section 4 for details).

Issue 3: Information Flow vs Acceptance

Besides the use of the modality $\mathbf{C}_A$, authority can also be delegated from one principal to another by nesting the says modality, as in the following statement from Example 1, which delegates the formula $read\_file1$ from principal $Admin$ to principal Bob:

2. $Admin \textsf{ says } ((Bob \textsf{ says } read\_file1) \rightarrow read\_file1)$

Intuitively, we expect (as in Example 1) that this formula together with Bob says $read\_file1$ should imply that $Admin$ says $read\_file1$. However, performing this inference requires us to infer from Bob says $read\_file1$ that $Admin$ says Bob says $read\_file1$. To allow for this inference, most authorization logics include the following axiom, or a stronger axiom that implies it (this axiom was proposed by Abadi [1]):

$$A \textsf{ says } \varphi \rightarrow B \textsf{ says } (A \textsf{ says } \varphi) \qquad \text{(I-SS)}$$

However, this axiom also allows *unwanted* statements to flow from one principal to another. Here is an example. Suppose $Admin$ delegates to Bob the authority to $read\_file1$ through statement (2), under the conception that Bob will only allow $read\_file1$ under reasonable conditions. However, Bob, mistakenly or maliciously, adds the following rule:

$$Bob \textsf{ says } (bad\_condition \rightarrow read\_file1)$$

where $bad\_condition$ means that a certain bad condition (for reading $file1$) holds. Now, using the statements above and (I-SS), Bob says $bad\_condition$ implies that $Admin$ says $read\_file1$, which is undesirable.

The problem here is that the logic, so far, does not provide a construct to allow $Admin$ to represent in statement (2) that it actually *trusts* the assumption (Bob says $read\_file1$). We propose to rectify this situation by including the construct A $\mathsf{ratified}$ φ, which means that A says φ and that this statement is trusted by the principal in the enclosing scope. With this construct, $Admin$ can revise its statement to say that:

2a. $Admin$ says $((Bob$ $\mathsf{ratified}$ $read\_file1) \to read\_file1)$

If Bob merely says $read\_file1$, it will imply $Admin$ says Bob says $read\_file1$, but not $Admin$ says Bob $\mathsf{ratified}$ $read\_file1$, and not allow for the deduction of $Admin$ says $read\_file1$. To allow for the latter, $Admin$ must make explicit rules to convert Bob's statements to ratified statements, e.g., it may add the following two rules:

3. $Admin$ says $((Bob$ says $good\_condition) \to (Bob$ $\mathsf{ratified}$ $good\_condition))$
4. $Admin$ says $((Bob$ says $(good\_condition \to read\_file1)) \to (Bob$ $\mathsf{ratified}$ $(good\_condition \to read\_file1)))$

thus allowing deduction of $Admin$ says $read\_file1$ from the statements Bob says $(good\_condition \to read\_file1)$ and Bob says $good\_condition$, but not from Bob says $(bad\_condition \to read\_file1)$ and Bob says $bad\_condition$. The formal rules that allow these deductions and a more detailed example of the use of ratification are presented in Section 4.

4 The New Modalities

In this section we present ACL$^+$, an access control logic with the modalities $\mathbf{P}_A$, $\mathbf{C}_A$ and A $\mathsf{ratified}$ $\bullet$. To summarize,

1. Permission and control can be represented directly in ACL$^+$ using the modalities $\mathbf{P}_A\varphi$ (principal A is authorized (permitted) φ) and $\mathbf{C}_A\varphi$ (principal A controls φ).
2. ACL$^+$ contains the operator A $\mathsf{ratified}$ φ, which means that principal A states φ and this statement has been ratified (or, is trusted) by the principal in whose context the formula is interpreted.

We introduce ACL$^+$ piecewise, starting with a simple access control logic containing the modality says defined by the following rules and axioms:

all axioms of intuitionistic propositional logic (IPC)

$$\frac{\vdash \varphi \quad \vdash \varphi \to \psi}{\vdash \psi} \quad \text{(MP)}$$

$$\frac{\vdash \varphi}{\vdash A \ \mathsf{says}\ \varphi} \quad \text{(nec-S)}$$

$$\vdash (A\ \mathsf{says}\ (\varphi \to \psi)) \to (A\ \mathsf{says}\ \varphi) \to (A\ \mathsf{says}\ \psi) \quad \text{(K-S)}$$

$$\vdash A\ \mathsf{says}\ \varphi \to B\ \mathsf{says}\ (A\ \mathsf{says}\ \varphi) \quad \text{(I-SS)}$$

We note that our logic is intuitionistic (constructive). The use of intuitionistic logic for access control has been motivated in prior work [12]; briefly, constructivism disallows proofs by contradiction, thus eliminating authorization if it is merely not denied. Axioms (MP) and (nec-S) express that says is a normal necessitation modality and are standard in access control literature.

4.1 Permission and Control

To this basic logic we add the modalities $\mathbf{P}_A$ and $\mathbf{C}_A$, characterized by the following rules and axioms:

Definition 1 (Axioms and Rules for $\mathbf{P}_A$ and $\mathbf{C}_A$).

$$\frac{\vdash \varphi}{\vdash \mathbf{C}_A \varphi} \tag{nec-C}$$

$$\vdash \mathbf{C}_A(\varphi \to \psi) \to (\mathbf{C}_A \varphi \to \mathbf{C}_A \psi) \tag{C-Deduce}$$

$$\vdash \mathbf{C}_A \varphi \to \mathbf{P}_A \varphi \tag{C2P}$$

$$\vdash \mathbf{P}_A(\varphi \vee \psi) \to \mathbf{P}_A \varphi \vee \mathbf{P}_A \psi \tag{or-P}$$

$$\vdash (\mathbf{C}_A \varphi \wedge (A \textsf{ says } \mathbf{C}_B \varphi)) \to \mathbf{C}_B \varphi \tag{del-C}$$

Axiom (C-Deduce) expresses that control is closed under logical deduction while rule (nec-C) means that all valid formulas of the logic are controlled by every principal A. Together, (nec-C) and (C-Deduce) make $\mathbf{C}_A$ a normal necessitation modality (similar to $\Box$ in standard modal logics). As motivated in Section 3, we model permission with a *possibility* modality, i.e., it is not closed under logical consequence, but we require it to distribute over disjunction (or-P). Axiom (C2P) relates the notion of control with that of permission and reads: "If principal A controls φ, then it is authorized (permitted) on φ". This implies that control of a formula is stronger than permission on the formula. Axiom (del-C) allows a principal A in control of φ to delegate that control to another principal B (see Example 2).

Definition 2 (Authorization). *Given a policy Γ, we say that A is* authorized *on access φ if and only if $\Gamma \vdash \mathbf{P}_A \varphi$.*

Example 2. The policy of Example 1 can be re-represented with the new modalities as follows

(1) $\mathbf{C}_{Admin}(read\_file1)$
(2) $Admin$ says $(\mathbf{C}_{Bob}(read\_file1))$

From (del-C), (MP) and (C2P) we can prove that Bob is authorized to read file1, i.e., $(1), (2) \vdash \mathbf{P}_{Bob}(read\_file1)$.

Example 3. A principal can selectively delegate privileges it controls to other principals. Consider a policy in which A controls the deletion of files 1 and 2. A can delegate to B only the authority to delete file 1 by asserting that B controls it. Formally,

$$\mathbf{C}_A(delete\_file1 \wedge delete\_file2), A \textsf{ says } (\mathbf{C}_B(delete\_file1)) \vdash \mathbf{C}_B(delete\_file1)$$

Proof. From the assumption $\mathbf{C}_A(delete\_file1 \wedge delete\_file2)$ infer using (nec-C) and (C-deduce) that $\mathbf{C}_A(delete\_file1)$. $\mathbf{C}_B(delete\_file1)$ follows using (del-C) and the assumption A says $(\mathbf{C}_B(delete\_file1))$.

4.2 The Modality (A ratified φ)

Next, we add to our logic the modality A ratified φ, which means not only that A says φ, but also that the latter has been checked, ratified, or is trusted by the principal in whose scope it occurs (Section 3)[2]. For instance, the formula B says (A ratified φ) means that: "A says φ and B ratified (trusts) this statement". The resulting logic is called ACL^+.

Like $\mathbf{C}_A$ and A says $\bullet$, we model A ratified $\bullet$ as a normal modality:

$$\frac{\vdash \varphi}{\vdash A \text{ ratified } \varphi} \tag{nec-R}$$

$$\vdash (A \text{ ratified } (\varphi \to \psi)) \to (A \text{ ratified } \varphi) \to (A \text{ ratified } \psi) \tag{K-R}$$

Further, the modality A ratified φ implies A says φ, but the converse is not true in general:

$$\vdash (A \text{ ratified } \varphi) \to (A \text{ says } \varphi) \tag{RS}$$

The axiom (RS) makes A ratified φ stronger than A says φ. Statement φ directly signed by a principal can be taken as an evidence of A says φ, not A ratified φ.

(I-SS) and (RS) together imply that:

$$\vdash (A \text{ ratified } \varphi) \to B \text{ says } A \text{ says } \varphi$$

but it is not possible to derive in general that

$$(A \text{ says } \varphi) \to B \text{ says } A \text{ ratified } \varphi$$

which would be unjustified because if A says φ, then B has not necessarily ratified it.

Example 4. The purpose of introducing ratified is to allow a principal control over what statements and proofs of another principal it will admit as trusted. Assume that a hospital administrator PA controls access to sensitive patient records. The main policy is that "a doctor has access to all patient records" and the determination of who constitutes a doctor comes from the principal HR, representing the human resources database. Let $\mathbf{C}_A(access\_records)$ mean that principal A has control over the access to patient records and $isDoctor\_A$ mean that A is a doctor. The main policy can be encoded as the formula[3]:

$$PA \text{ says } \bigwedge_{A\in\mathcal{P}}[(HR \text{ ratified } isDoctor\_A) \to (\mathbf{C}_A(access\_records))] \tag{P1}$$

Observe that we are using (HR ratified ...) inside the policy instead of (HR says ...) to make sure that consequences of the policy depend only on statements of HR that have been ratified by PA.

Now, PA can choose to trust the policies of HR selectively. For instance, if PA trusts all deductions of the form $isDoctor\_A$ that HR may make, it can have the policy:

[2] The idea of interpreting formulas in the context of principals goes back to the logics DTL and BL [10].

[3] Because we are using a propositional language, we assume principals to range over a *finite* set $\mathcal{P}$. Accordingly, $\bigwedge_{A\in\mathcal{P}} \varphi$ reads "For all principals A in $\mathcal{P}$, φ holds".

$$PA \text{ says } \bigwedge_{A\in\mathcal{P}}[(HR \text{ says } \mathit{isDoctor\_A}) \rightarrow (HR \text{ ratified } \mathit{isDoctor\_A})] \quad \text{(P2)}$$

Then, for any principal A, we have that

$$(P1), (P2), HR \text{ says } (\mathit{isDoctor\_A}) \vdash PA \text{ says } \mathbf{C}_A(\mathit{access\_records})$$

If, on the other hand, PA only trusts HR's statements about two principals $Alice$ and Bob, it can selectively assert (in place of (P2)) that:

$$PA \text{ says } ((HR \text{ says } \mathit{isDoctor\_Alice}) \rightarrow (HR \text{ ratified } \mathit{isDoctor\_Alice}))$$
$$PA \text{ says } ((HR \text{ says } \mathit{isDoctor\_Bob}) \rightarrow (HR \text{ ratified } \mathit{isDoctor\_Bob}))$$

As a last illustration, suppose that the HR has two policies, one of which states that every administrator is a doctor and the other of which (mistakenly) states that every hospital employee is a doctor:

$$HR \text{ says } \bigwedge_{A\in\mathcal{P}}(\mathit{isAdmin\_A} \rightarrow \mathit{isDoctor\_A}) \quad \text{(P3)}$$
$$HR \text{ says } \bigwedge_{A\in\mathcal{P}}(\mathit{isEmployee\_A} \rightarrow \mathit{isDoctor\_A}) \quad \text{(P4)}$$

PA can choose to ratify the first of these, but not the second, by asserting in place of (P2) that:

$$PA \text{ says } ((HR \text{ says } \bigwedge_{A\in\mathcal{P}}(\mathit{isAdmin\_A} \rightarrow \mathit{isDoctor\_A})) \rightarrow (HR \text{ ratified } \bigwedge_{A\in\mathcal{P}}(\mathit{isAdmin\_A}) \rightarrow \mathit{isDoctor\_A}))) \quad \text{(P5)}$$

$$PA \text{ says } \bigwedge_{A\in\mathcal{P}}((HR \text{ says } \mathit{isAdmin\_A}) \rightarrow (HR \text{ ratified } \mathit{isAdmin\_A})) \quad \text{(P6)}$$

Suppose that HR says $\mathit{isAdmin\_Alice}$. Then, we can deduce PA says $\mathbf{C}_{Alice}(\mathit{access\_records})$ from (P1), (P3), (P5) and (P6) as follows:

1. From (P3) and (I-SS), deduce that

$$PA \text{ says } (HR \text{ says } (\bigwedge_{A\in\mathcal{P}} (\mathit{isAdmin\_A} \rightarrow \mathit{isDoctor\_A})))$$

2. From (1), (K-S) and (P5) deduce that

$$PA \text{ says } (HR \text{ ratified } (\bigwedge_{A\in\mathcal{P}} (\mathit{isAdmin\_A} \rightarrow \mathit{isDoctor\_A})))$$

3. From (HR says $\mathit{isAdmin\_Alice}$) and (I-SS) deduce that (PA says HR says $\mathit{isAdmin\_Alice}$)
4. From (3), (K-S) and (P6) deduce that (PA says HR ratified $\mathit{isAdmin\_Alice}$)
5. From (2), (4), (K-S), and (K-R) deduce that (PA says HR ratified $\mathit{isDoctor\_Alice}$)
6. From (5), (P1), and (K-S) deduce that (PA says $\mathbf{C}_{Alice}(\mathit{access\_records})$)

If we replace the assumption (HR says $\mathit{isAdmin\_Alice}$) with the assumption (HR says $\mathit{isEmployee\_Alice}$), then we cannot deduce (PA says ($\mathbf{C}_{Alice}(\mathit{access\_records})$)) because we cannot deduce (5) above. In place of (5), we can deduce only the weaker statement (PA says (HR says $\mathit{isDoctor\_Alice}$)), which does not imply (PA says $\mathbf{C}_{Alice}(\mathit{access\_records})$) in our theory.

5 Semantics

In this section, we define sound and complete semantics for ACL$^+$. Our semantics uses graph-based structures called Kripke models, that are standard in modal logic. Although Kripke semantics are not necessarily intuitive, they lead directly to a proof theory for the logic, a decidability result for it, and an implementation of its decision procedure (Section 6).

Definition 3. *An intuitionistic model, $\mathcal{M}$, of ACL$^+$ is a tuple*

$$(W, \leq, \{S_A\}_{A\in\mathcal{P}}, \{C_A\}_{A\in\mathcal{P}}, \{R_A\}_{A\in\mathcal{P}}, \{P_A\}_{A\in\mathcal{P}}, h)$$

where

- *$\mathcal{P}$ is a set of principals.*
- *$(W, \leq)$ is a preorder, where elements of W are called states or worlds, and $\leq$ is a binary relation over W which satisfies the following conditions*
  $\forall x.(x \leq x)$ (refl)
  $\forall x, y, z.((x \leq y) \wedge (y \leq z) \rightarrow (x \leq z))$ (trans)
- *S_A, C_A, R_A and P_A are binary relations on W that satisfy the following conditions:*
  $\forall x, y, z, w.(((x \leq y) \wedge (yS_Az) \wedge (z \leq w)) \rightarrow (xS_Aw))$ (mon-S)
  $\forall x, y, z, w.(((x \leq y) \wedge (yC_Az) \wedge (z \leq w)) \rightarrow (xC_Aw))$ (mon-C)
  $\forall x, y, z, w.(((x \leq y) \wedge (yR_Az) \wedge (z \leq w)) \rightarrow (xR_Aw))$ (mon-R)
  $\forall x, y, z, w.(((x \leq y) \wedge (zP_Ay) \wedge (z \leq w)) \rightarrow (wP_Ax))$ (mon-P)
- *h is an assignment which, for each atom q, assigns the subset of worlds $h(q) \subseteq W$ where q holds. Moreover, we require h to be monotone w.r.t. $\leq$, i.e., if $x \in h(q)$ and $x \leq y$ then $y \in h(q)$.*

Conditions above ensure monotonicity *of the logic (Lemma 1), which is a standard property of Kripke semantics for constructive logics. Moreover, to force ACL$^+$ models to admit the axioms (I-SS), (C2P), (del-C) and (RS) we require the following to hold for any two principals A and B.*

$\forall x, y, z.(((xS_By) \wedge (yS_Az)) \rightarrow (xS_Az))$ (s-I-SS)
$\forall x \exists y.(xC_Ay \wedge xP_Ay)$ (s-C2P)
$\forall x, y.((xC_By) \rightarrow ((xC_Ay) \vee \exists z((xS_Az) \wedge (zC_By)))$ (s-del-C)
$\forall x, y.((xS_Ay) \rightarrow (xR_Ay))$ (s-RS)

An *interpretation* for the logic is a pair $\mathcal{M}, t$ where $\mathcal{M}$ is a model and t is a world in $\mathcal{M}$.

Definition 4 (Satisfaction Relation). *The satisfaction relation "$\models$" between interpretations and formulae of the logic is defined as follows.*

- $\mathcal{M}, t \models q$ iff $t \in h(q)$
- $\mathcal{M}, t \not\models \bot$
- $\mathcal{M}, t \models \varphi \vee \psi$ iff $\mathcal{M}, t \models \varphi$ or $\mathcal{M}, t \models \psi$
- $\mathcal{M}, t \models \varphi \wedge \psi$ iff $\mathcal{M}, t \models \varphi$ and $\mathcal{M}, t \models \psi$

- $\mathcal{M}, t \models \varphi \rightarrow \psi$ iff for all $s, t \leq s$ and $\mathcal{M}, s \models \varphi$ implies $\mathcal{M}, s \models \psi$
- $\mathcal{M}, t \models \neg\varphi$ iff for all $s, t \leq s$ implies $\mathcal{M}, t \not\models \varphi$
- $\mathcal{M}, t \models A$ says φ iff for all s such that $tS_A s$ we have $\mathcal{M}, s \models \varphi$
- $\mathcal{M}, t \models \mathbf{C}_A\varphi$ iff for all s such that $tC_A s$ we have $\mathcal{M}, s \models \varphi$
- $\mathcal{M}, t \models A$ ratified φ iff for all s such that $tR_A s$ we have $\mathcal{M}, s \models \varphi$
- $\mathcal{M}, t \models \mathbf{P}_A\varphi$ iff there exists an s, $tP_A s$ such that $\mathcal{M}, s \models \varphi$

Lemma 1 (Monotonicity). *For any formula φ and any interpretation $\mathcal{M}, t$, if $\mathcal{M}, t \models \varphi$ and $t \leq s$ then $\mathcal{M}, s \models \varphi$.*

We say that $\mathcal{M} \models \varphi$ if for all $t \in \mathcal{M}$, it is the case that $\mathcal{M}, t \models \varphi$. Further, $\Gamma \models \varphi$ if for every $\mathcal{M}$ satisfying the above conditions, $\mathcal{M} \models \Gamma$ implies $\mathcal{M} \models \varphi$.

Theorem 1 (Soundness). *If $\vdash \varphi$ then $\models \varphi$*

Theorem 2 (Completeness). *If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$*

We note that the conditions (s-I-SS), (s-C2P), (s-del-C) and (s-RS) are *canonical* for the axioms (I-SS), (C2P), (del-C) and (RS), respectively, i.e., a logic with any subset of these axioms is sound and complete with respect to models that satisfy the conditions corresponding to the chosen axioms.

6 A Semantic-Based Calculus for ACL$^+$

In this section we briefly present Seq-ACL$^+$, a sound, complete and cut-free sequent calculus for ACL$^+$. The calculus is inspired by the work of Negri [19][4] and follows the so-called labeled approach [4,20], which directly uses the Kripke semantics. The use of labeled sequent calculi for access control is relatively new and has been introduced in [15,16] to define proof theory of a specific says-based access control logic. The sequent calculus directly leads to a decision procedure for the logic ACL$^+$.

Seq-ACL$^+$ manipulates two types of labeled formulas:

1. *World formulas*, denoted by $x : \varphi$, where x is a world and φ is a well formed formula of ACL$^+$, intuitively meaning that φ holds in world x.
2. *Transition formulas* representing semantic accessibility relationships. These formulas have one of the forms $xS_Ay, xC_Ay, xR_Ay, xP_Ay$ and $x \leq y$.

A sequent is a tuple $\langle \Sigma, \mathbb{M}, \Gamma, \Delta \rangle$, usually written $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ where $\mathbb{M}, \Gamma$ and Δ are multisets of labeled formulas and Σ is the set of labels (worlds) appearing in the rest of the sequent. Intuitively, the sequent $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ means that "every model which satisfies all labeled formulas of $\Gamma \cup \mathbb{M}$ satisfies at least one labeled formula in Δ". This is made precise by the notion of *validity* in the following definition.

Definition 5 (Sequent validity). *Given a model*

$$\mathcal{M} = (W, \leq, \{S_A\}_{A \in \mathcal{P}}, \{C_A\}_{A \in \mathcal{P}}, \{R_A\}_{A \in \mathcal{P}}, \{P_A\}_{A \in \mathcal{P}}, h)$$

and a label alphabet $\mathcal{A}$, consider a mapping $I : \mathcal{A} \rightarrow W$. Let F denote a labeled formula. Define $\mathcal{M} \models_I F$ as follows:

[4] In particular, proofs of metatheorems about Seq-ACL$^+$ use methods developed in [19].

- $\mathcal{M} \models_I x : \alpha$ *iff* $\mathcal{M}, I(x) \models \alpha$
- $\mathcal{M} \models_I xC_A y$ *iff* $I(x) C_A I(y)$ *(Similarly for* S_A, R_A, P_A *and* $\leq$*).*

We say that $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ *is valid in* $\mathcal{M}$ *if, for every mapping* $I : \mathcal{A} \to W$*, if* $\mathcal{M} \models_I F$ *for every* $F \in \mathbb{M} \cup \Gamma$*, then* $\mathcal{M} \models_I G$ *for some* $G \in \Delta$*. We say that* $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ *is valid in Seq-ACL*$^+$ *if it is valid in every* $\mathcal{M}$*.*

Figure 1 lists the rules of the calculus Seq-ACL$^+$, divided into four groups.

- *Axiom rules* do not have premises and describe valid sequents.
- *Logical rules* operate on connectives of the logic. For reasons of space, we omit some rules: ∧L,∧R,∨L and ∨R[5] are standard (see, for instance, [19]) while rules for says and ratified have the same structure as those for **C**.
- *Semantic rules* define the properties that hold for relationships $\leq, S_A, R_A, C_A$ and P_A in all ACL$^+$ models.
- *Access control rules* codify axioms that differentiate ACL$^+$ from other constructive normal modal logics, i.e., (I-SS), (C2P), (del-C) and (RS).

Note that semantic and access control rules are in one-to-one correspondence with semantic conditions of Definition 3.

We say that a sequent $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ is *derivable* in Seq-ACL$^+$ if it admits a *derivation*. A derivation is a tree whose nodes are sequents. A branch is a sequence of nodes $\Sigma_1; \mathbb{M}_1; \Gamma_1 \Rightarrow \Delta_1, \Sigma_2; \mathbb{M}_2; \Gamma_2 \Rightarrow \Delta_2, \ldots, \Sigma_n; \mathbb{M}_n; \Gamma_n \Rightarrow \Delta_n, \ldots$ Each node $\Sigma_i; \mathbb{M}_i; \Gamma_i \Rightarrow \Delta_i$ is obtained from its immediate successor $\Sigma_{i-1}; \mathbb{M}_{i-1}; \Gamma_{i-1} \Rightarrow \Delta_{i-1}$ by applying *backward* a rule of Seq-ACL$^+$, having $\Sigma_{i-1}; \mathbb{M}_{i-1}; \Gamma_{i-1} \Rightarrow \Delta_{i-1}$ as the conclusion and $\Sigma_i; \mathbb{M}_i; \Gamma_i \Rightarrow \Delta_i$ as one of its premises. A branch is closed if one of its nodes is an instance of axiom rules, otherwise it is open. We say that a tree is closed if all of its branches are closed. A sequent $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ has a derivation in Seq-ACL$^+$ if there is a closed tree having $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ as the root. As an example we show a derivation of the axiom (C2P)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x,y,z; x \leq y, z \leq z, yC_Az, yP_Az; y : \mathbf{C}_A p, z : p \Rightarrow y : \mathbf{P}_A p, z : p}\,\text{init}}{x,y,z; x \leq y, yC_Az, yP_Az; y : \mathbf{C}_A p, z : p \Rightarrow y : \mathbf{P}_A p, z : p}\,\text{refl}}{x,y,z; x \leq y, yC_Az, yP_Az; y : \mathbf{C}_A p, z : p \Rightarrow y : \mathbf{P}_A p}\,\mathbf{PR}}{x,y,z; x \leq y, yC_Az, yP_Az; y : \mathbf{C}_A p \Rightarrow y : \mathbf{P}_A p}\,\mathbf{CL}}{x,y; x \leq y; y : \mathbf{C}_A p \Rightarrow y : \mathbf{P}_A p}\,\text{s-C2P}}{x;\ ; \Rightarrow x : \mathbf{C}_A p \to \mathbf{P}_A p}\,{\to}\text{R}$$

Theorem 3 (Admissibility of cut). *$\Sigma; \mathbb{M}; \Gamma \Rightarrow x : \alpha, \Delta$ and $\Sigma; \mathbb{M}; \Gamma, x : \alpha \Rightarrow \Delta$ imply $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$.*

Theorem 4 (Soundness of Seq-ACL$^+$). *If a sequent $\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta$ is derivable then it is valid in the sense of Definition 5.*

Theorem 5 (Completeness of Seq-ACL$^+$). *If a formula α is valid in ACL$^+$ (i.e., $\models \alpha$), then $x; ; \Rightarrow x : \alpha$ is derivable in Seq-ACL$^+$.*

[5] We do not have specific rules for negation because, in constructive logics, $\neg\varphi$ is equivalent to $\varphi \to \bot$.

Axiom Rules

$$\frac{}{\Sigma; \mathbb{M}, x \leq y; \Gamma, x : p \Rightarrow y : p, \Delta}\text{init} \qquad \frac{}{\Sigma; \mathbb{M}; \Gamma, x : \bot \Rightarrow \Delta}\bot\text{L} \qquad \frac{}{\Sigma; \mathbb{M}; \Gamma \Rightarrow x : \top, \Delta}\top\text{R}$$

Logical Rules

$$\frac{\Sigma; \mathbb{M}, x \leq y; \Gamma, x : \alpha \to \beta \Rightarrow_{\mathcal{T}} y : \alpha, \Delta \quad \Sigma; \mathbb{M}, x \leq y; \Gamma, x : \alpha \to \beta, y : \beta \Rightarrow_{\mathcal{T}} \Delta}{\Sigma; \mathbb{M}, x \leq y; \Gamma, x : \alpha \to \beta \Rightarrow \Delta}\to\text{L}$$

$$\frac{\Sigma, y; \mathbb{M}, x \leq y; \Gamma, y : \alpha \Rightarrow y : \beta, \Delta}{\Sigma; \mathbb{M}; \Gamma \Rightarrow x : \alpha \to \beta, \Delta}\underset{y\text{ new}}{\to\text{R}}$$

$$\frac{\Sigma; \mathbb{M}, xC_Ay; \Gamma, x : \mathbf{C}_A\alpha, y : \alpha \Rightarrow \Delta}{\Sigma; \mathbb{M}, xC_Ay; \Gamma, x : \mathbf{C}_A\alpha \Rightarrow \Delta}\mathbf{C}\text{L} \qquad \frac{\Sigma, y; \mathbb{M}, xC_Ay; \Gamma \Rightarrow y : \alpha, \Delta}{\Sigma; \mathbb{M}; \Gamma \Rightarrow x : \mathbf{C}_A\alpha, \Delta}\underset{y\text{ new}}{\mathbf{C}\text{R}}$$

$$\frac{\Sigma; \mathbb{M}, xP_Ay; \Gamma \Rightarrow x : \mathbf{P}_A\alpha, y : \alpha, \Delta}{\Sigma; \mathbb{M}, xP_Ay; \Gamma \Rightarrow x : \mathbf{P}_A\alpha, \Delta}\mathbf{P}\text{R} \qquad \frac{\Sigma, y; \mathbb{M}, xP_Ay; \Gamma, y : \alpha \Rightarrow \Delta}{\Sigma; \mathbb{M}; \Gamma, x : \mathbf{P}_A\alpha \Rightarrow \Delta}\underset{y\text{ new}}{\mathbf{P}\text{L}}$$

Semantical Rules

$$\frac{\Sigma; \mathbb{M}, x \leq y, yS_Az, z \leq w, xS_Aw; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, x \leq y, yS_Az, z \leq w; \Gamma \Rightarrow \Delta}\text{mon-S} \quad \frac{\Sigma; \mathbb{M}, x \leq y, yC_Az, z \leq w, xC_Aw; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, x \leq y, yC_Az, z \leq w; \Gamma \Rightarrow \Delta}\text{mon-C}$$

$$\frac{\Sigma; \mathbb{M}, x \leq y, yR_Az, z \leq w, xR_Aw; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, x \leq y, yR_Az, z \leq w; \Gamma \Rightarrow \Delta}\text{mon-R} \quad \frac{\Sigma; \mathbb{M}, x \leq y, zP_Ay, z \leq w, wP_Ax; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, x \leq y, zP_Ay, z \leq w; \Gamma \Rightarrow \Delta}\text{mon-P}$$

$$\frac{\Sigma; \mathbb{M}, x \leq x; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta}\underset{x \in \Sigma}{\text{refl}} \qquad \frac{\Sigma; \mathbb{M}, x \leq y, y \leq z, x \leq z; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, x \leq y, y \leq z; \Gamma \Rightarrow \Delta}\text{trans}$$

Access Control Rules

$$\frac{\Sigma; \mathbb{M}, xS_By, yS_Az, xS_Az; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, xS_By, yS_Az; \Gamma \Rightarrow \Delta}\text{s-I-SS} \qquad \frac{\Sigma, y; \mathbb{M}, xC_Ay, xP_Ay; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}; \Gamma \Rightarrow \Delta}\underset{y\text{ new}}{\text{s-C2P}}$$

$$\frac{\Sigma; \mathbb{M}, xC_By, xC_Ay; \Gamma \Rightarrow \Delta \quad \Sigma, z; \mathbb{M}, xC_By, xS_Az, zC_By; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, xC_By; \Gamma \Rightarrow \Delta}\underset{z\text{ new}}{\text{s-del-C}}$$

$$\frac{\Sigma; \mathbb{M}, xS_Ay, xR_Ay; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, xS_Ay; \Gamma \Rightarrow \Delta}\text{s-RS}$$

Fig. 1. Seq-ACL$^+$ Rules

6.1 Termination

Next, we use the sequent calculus Seq-ACL$^+$ to prove that the logic ACL$^+$ is decidable. Note that admissibility of cut (Theorem 3) alone does not ensure the termination of backward proof search in the sequent calculus because access control rules and the rules saysL, $\mathsf{ratifiedL}$, $\mathbf{C}$L, and $\mathbf{P}$R may increase the complexity of sequents in a backward proof search. Accordingly, we prove that these "critical" rules can be applied in a controlled way. For instance, the following Lemma states the use of $\mathbf{C}$L, $\mathbf{P}$R, and access control rules can be limited. (Without loss of generality we assume that the root of each proof has the form $x; ; \Rightarrow x : \varphi$).

Lemma 2 (Controlled use of rules). *In each branch of a backward proof search, it is useless to: (1) apply $\mathbf{C}L$ on the same transition relation $xC_Ay \in \mathbb{M}$ more than once, (2) apply $\mathbf{P}R$ on the same transition relation $xP_Ay \in \mathbb{M}$ more than once, (3) apply rule χ for $\chi \in \{$mon-S,mon-R,mon-C,mon-P,sym,trans,s-I-SS,s-del-C,s-C2P,s-RS$\}$ on the same transition formula (or label as in s-RS) more than once.*

However, even the above Lemma is not sufficient to ensure termination of backward proof search. In particular, there are two issues:

1. Interaction of the rule (trans) with $\rightarrow L$ adds new accessible worlds, and we can build chains of accessible worlds on which $\rightarrow L$ can be applied *ad infinitum*.
2. Application of rules (s-del-C) and (s-C2P) generates transition relations with new labels that can be used for repeated application of the same rules.

We bound the number of such interactions using a counting argument. Let $depth(F)$ be the height of the parse tree of formula F.

Definition 6 (Label distance). *Given a sequent $\Sigma, \mathbb{M}, \Gamma \Rightarrow \Delta$ and two labels x and y such that $x \leq y \in \mathbb{M}$, we define the distance $d(x, y)$ between two labels as 0 when $x = y$ and n when $x \neq y$, where n is the length of the* longest *sequence of transitions in $\mathbb{M}$ "connecting" the two labels, i.e., $x \overset{\sim}{\bigcirc} x_1, x_1 \overset{\sim}{\bigcirc} x_2, \ldots, x_{n-1} \overset{\sim}{\bigcirc} y$ where $\overset{\sim}{\bigcirc} \in \{S_A, C_A, R_A, P_A, \leq\}$ (for any principal A). As an example if $\{x \leq y, yC_Az, zP_Ak, xS_Ak\} \in \mathbb{M}$ we have $d(x, k) = 3$.*

Lemma 3 (Bounded application of rules). *Let x, x_1 be labels and F a formula such that $d(x, x_1) > depth(F)$. Then, in each branch of a backwards proof search starting with $x; ; \Rightarrow x : F$, it is useless to: (1) apply →L on a transition formula $x_1 \leq x_2$, (2) apply s-C2P on a label x_1, (3) apply s-del-C on a transition formula $x_1 C_B x_2$.*

Using this lemma, we obtain decidability for ACL$^+$.

Theorem 6 (Decidability). *The logic ACL$^+$ is decidable.*

Our proof of decidability directly leads to a decision procedure for ACL$^+$. A Prolog implementation of the procedure is available from the authors' webpages.

$$\frac{\Sigma; \mathbb{M}, xS_Ay, xS_By; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, xS_By; \Gamma \Rightarrow \Delta}\text{s-sub-S}^A_B \qquad \frac{\Sigma; \mathbb{M}, xR_Ay, xR_By; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, xR_By; \Gamma \Rightarrow \Delta}\text{s-sub-R}^A_B$$

$$\frac{\Sigma; \mathbb{M}, xP_Ay, xP_By; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, xP_Ay; \Gamma \Rightarrow \Delta}\text{s-sub-P}^A_B \qquad \frac{\Sigma; \mathbb{M}, xC_By, xC_Ay; \Gamma \Rightarrow \Delta}{\Sigma; \mathbb{M}, xC_By; \Gamma \Rightarrow \Delta}\text{s-sub-C}^A_B$$

Fig. 2. Access Control Rules for Subordination

7 Extending Seq-ACL$^+$ with Constructs for Subordination

The correspondence between semantic conditions and axioms allows us to modularly extend ACL$^+$ with new axioms, and new (corresponding) sequent calculus rules. The difficult aspect in any such extension is to prove decidability. As a specific case, we show here how we may extend the logic with new *subordination axioms* of any of the following forms, and obtain decidability again. (In these axioms A and B are specific principals, not metavariables, but φ is a metavariable standing for all formulas.)

$$\vdash A \text{ says } \varphi \rightarrow B \text{ says } \varphi \qquad \text{(sub-S)}^A_B$$
$$\vdash A \text{ ratified } \varphi \rightarrow B \text{ ratified } \varphi \qquad \text{(sub-R)}^A_B$$
$$\vdash \mathbf{P}_A\varphi \rightarrow \mathbf{P}_B\varphi \qquad \text{(sub-P)}^A_B$$
$$\vdash \mathbf{C}_A\varphi \rightarrow \mathbf{C}_B\varphi \qquad \text{(sub-C)}^A_B$$

We call these axioms subordination axioms because each axiom suggests that one of the two principals A and B is subordinate to the other. The first (second) axiom means that statements (ratifications) of A are echoed by B, so B is, in a sense, subordinate to A. The third (fourth) axiom means that if A has a permission (ability to control), then so does B, so B is more powerful than A.

Definition 7. *The semantic conditions on models corresponding to the axioms above are, respectively:*

$$\forall x, y.(xS_By \rightarrow xS_Ay) \qquad \textit{(s-sub-S)}^A_B$$
$$\forall x, y.(xR_By \rightarrow xR_Ay) \qquad \textit{(s-sub-R)}^A_B$$
$$\forall x, y.(xP_Ay \rightarrow xP_By) \qquad \textit{(s-sub-P)}^A_B$$
$$\forall x, y.(xC_By \rightarrow xC_Ay) \qquad \textit{(s-sub-C)}^A_B$$

Corresponding access control rules for the sequent calculus are shown in Figure 2.

Lemma 4. *The extension of Seq-ACL$^+$ with any subset of the rules in Figure 2 is sound and complete w.r.t. models that satisfy corresponding conditions from Definition 7. Further, the extended calculus admits cut and is decidable.*

8 Related Work

The study of formal properties of says and other constructs in modal logic is a relatively new research trend. Prior work by the second author [10] adopts a modified version of constructive modal logic S4 called DTL$_0$ and shows how existing access control logics can be embedded (via translation) into DTL$_0$. Other work [11] translates existing access control logics into S4 by relying on a slight simplification of Gödel's translation from intuitionistic logic to S4, and extending it to formulas of the form A says φ. The first author has developed conditional logics as a general framework for modular sequent calculi for standard access control logics with the says connective [15,16]. Dinesh et al. [9] present an access control logic based on says and extended with obligation and permissions, but their treatment of permissions is different from ours and is closely tied to says. The use of canonical properties for access control axioms was first considered in [8] where standard access control axioms (e.g. (unit) and (hand-off)) are characterized in terms of first-order conditions on Kripke models.

9 Conclusion

We have presented ACL$^+$, a constructive multi-modal logic for access control that introduces three new modalities $\mathbf{P}_A$ (permission), $\mathbf{C}_A$ (control), and ratified (trusted statement) to fix some practical problems in reasoning with policies. The connectives of the logic are defined by a sound and complete Kripke semantics for ACL$^+$ together with a correspondence between conditions on models and the logic's axioms. The semantics lead to Seq-ACL$^+$, a sound, complete, cut-free and terminating calculus for ACL$^+$. Finally, ACL$^+$ can be extended with new axioms, as illustrated by examples of axioms for specific kinds of subordination among principals.

Acknowledgments Valerio Genovese was supported by the National Research Fund, Luxembourg. Deepak Garg was supported by the U.S. Army Research Office contract "Perpetually Available and Secure Information Systems" (DAAD19-02-1-0389) to Carnegie Mellon CyLab and the AFOSR MURI "Collaborative Policies and Assured Information Sharing".

References

1. Abadi, M.: Logic in access control. In: Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS). pp. 228–233 (2003)
2. Abadi, M.: Variations in access control logic. In: Proceedings of the 9th International Conference on Deontic Logic in Computer Science (DEON). pp. 96–109 (2008)
3. Abadi, M.: Logic in access control (tutorial notes). In: Proceedings of the 9th International School on Foundations of Security Analysis and Design (FOSAD). pp. 145–165 (2009)
4. Basin, D., D'Agostino, M., Gabbay, D.M., Matthews, S., Viganó, L.: Labelled Deduction. Spinger (2000)
5. Bauer, L.: Access Control for the Web via Proof-Carrying Authorization. Ph.D. thesis, Princeton University (2003)
6. Bauer, L., Garriss, S., McCune, J.M., Reiter, M.K., Rouse, J., Rutenbar, P.: Device-enabled authorization in the Grey system. In: Proceedings of the 8th International Conference on Information Security (ISC). pp. 431–445 (2005)
7. Becker, M.Y., Fournet, C., Gordon, A.D.: SecPAL: Design and semantics of a decentralized authorization language. Journal of Computer Security 18(4), 619–665 (2010)
8. Boella, G., Gabbay, D.M., Genovese, V., van der Torre, L.: Fibred security language. Studia Logica 92(3), 395–436 (2009)

9. Dinesh, N., Joshi, A.K., Lee, I., Sokolsky, O.: Permission to speak: A logic for access control and conformance. Journal of Logic and Algebraic Programming 80(1), 50–74 (2011)
10. Garg, D.: Principal centric reasoning in constructive authorization logic. In: Informal Proceedings of Intuitionistic Modal Logic and Application (IMLA) (2008), Full version available as Carnegie Mellon Technical Report CMU-CS-09-120
11. Garg, D., Abadi, M.: A modal deconstruction of access control logics. In: Proceedings of the 11th International Conference on Foundations of Software Science and Computational Structures (FoSSaCS). pp. 216–230 (2008)
12. Garg, D., Pfenning, F.: Non-interference in constructive authorization logic. In: Proceedings of the 19th IEEE Computer Security Foundations Workshop (CSFW). pp. 283–293 (2006)
13. Garg, D., Pfenning, F.: A proof-carrying file system. In: Proceedings of the 31st IEEE Symposium on Security and Privacy (Oakland). pp. 349–364 (2010)
14. Genovese, V., Giordano, L., Gliozzi, V., Pozzato, G.L.: A constructive conditional logic for access control: a preliminary report. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI). pp. 1073–1074 (2010)
15. Genovese, V., Giordano, L., Gliozzi, V., Pozzato, G.L.: Logics for access control: A conditional approach. In: Informal Proceedings of the 1st Workshop on Logic in Security (LIS). pp. 78–92 (2010)
16. Genovese, V., Giordano, L., Gliozzi, V., Pozzato, G.L.: A conditional constructive logic for access control and its sequent calculus. In: Proceedings of the 20th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX) (2011), to appear
17. Gurevich, Y., Neeman, I.: Logic of infons: The propositional case. ACM Transactions on Computational Logic 12(2) (2011)
18. Lampson, B.W., Abadi, M., Burrows, M., Wobber, E.: Authentication in distributed systems: Theory and practice. ACM Transactions on Computer Systems 10(4), 265–310 (1992)
19. Negri, S.: Proof analysis in modal logic. Journal of Philosophical Logic 34, 507–544 (2005)
20. Negri, S., von Plato, J.: Proof Analysis. Cambridge University Press (2011)
21. Schneider, F.B., Walsh, K., Sirer, E.G.: Nexus Authorization Logic (NAL): Design rationale and applications. Tech. rep., Cornell University (2009), online at `http://ecommons.library.cornell.edu/handle/1813/13679`
22. Wobber, E., Abadi, M., Burrows, M.: Authentication in the taos operating system. ACM Transactions on Computer Systems 12(1), 3–32 (1994)