A Multimodal view on Access Control and Trust Management: Fibred Security Language

Guido Boella [1]

Dept. of Computer Science
Università di Torino
Torino, Italia

Dov M. Gabbay [2]

Dept. of Computer Science
King's College
London, UK

Valerio Genovese [3]

Dept. of Computer Science
Università di Torino
Torino, Italia

Leon van der Torre [4]

Dept. of Computer Science and Communications
University of Luxembourg
Luxebmourg City, Luxembroug

Abstract

We study access control policies based on the **says** operator by introducing a logical framework called Fibred Security Language (FSL) which is able to deal with features like joint responsibility between sets of principals and to identify them by means of first-order formulas. FSL is based on a multimodal logic methodology. We first discuss the main contributions from the expressiveness point of view, we give semantics for the language (both for classical and intuitionistic fragment), we then prove that in order to express well-known properties like 'speaks-for' or 'hand-off', defined in terms of **says**, we do not need second-order logic (unlike previous approaches) but a decidable fragment of first-order logic suffices. Finally we propose a model-driven study of the **says** axiomatization by constraining the Kripke models in order to respect desirable security properties.

Keywords: Access Control, Language-based Security, Trust Management

[1] Email: guido@di.unito.it
[2] Email: dov.gabbay@kcl.ac.uk
[3] Email: valerio.click@gmail.com
[4] Email: leon.vandertorre@uni.lu

1 Introduction

Access control is a pervasive issue in security: it consists in determining whether the principal (a key, channel, machine, user, program) that issues a request to access a resource should be trusted on its request, i.e., if it is authorized. Authorization can be based in the simplest case on access control lists (ACL) associated with resources or with capabilities held by principals, but it may be complicated, for instance, by membership of groups, roles and delegation. Thus, logics are often introduced in access control to express policies and to enable reasoning about principals and their requests, and other general statements.

Logical approaches to comprehend, analyse, create and verify the policies and control mechanisms used to protect resources have been extensively studied in these years [1,3,7,8,14,18]. The interest of the community is also underlined by a number of research projects that have applied these logics for designing or motivating various languages and systems [4,6,9,19]. On the other hand, as reported in [12], there have been only few, limited efforts to study the logics themselves (e.g. [1,3,13]). In particular, the problems of the axiomatization of the well-known **says** operator have been studied only recently in [12,2]. Generally, the full expressiveness of the proposed logics is reached by exploiting second-order formalisms in order to axiomatize important concepts like the speaks-for or hand-off [3,16]. In this paper we introduce a novel first-order multimodal logic for access control in distributed systems called Fibred Security Language (FSL). The contribution of FSL addresses the following research questions:

(i) How to define a general language capable to embody and extend existing access control logics?

(ii) How to formalize a logic which provides axiomatization of security properties that avoids undesired side effects and which at the same time ensures tractability?

Our methodology is centered on a first-order language based on the fibring approach of Gabbay [11] and goes in the direction of having a method to integrate different logics into a single system.

We use a multimodal approach to express axioms which in the literature have been expressed in (computationally intractable) second-order logics within a first-order logic. The reduction from second-order to first-order can be of practical value in the objective of building theorem provers in support of proof-carrying authorization mechanisms.

In this paper we focus on the expressiveness of the logic and on the advantages in the employment of a fibred multimodal semantics. There is no space to provide a calculus for automated theorem proving and to translate existing approaches into FSL.

In Section 2 we briefly present FSL by showing its expressiveness. In Section 3 we list and comment the axioms of the says operator defined in [2]. Section 4 is devoted to the introduction and formalization of the basic system FSL. In Section 5 we show how the second-order axioms introduced in Section 3 can be translated into first-order constraints on the multimodal-kripke semantics of FSL. Section 6

ends the paper.

2 The Fibred Security Language

From the expressivity point of view, FSL aims to extend previous logics for access control by introducing joint responsibility between principals and groups of principals as first-class citizen described by means of first-order formulas. Although these properties are employed in general languages to describe policies [5], FSL is the first *logical* approach which embodies these features with a coherent semantical formalization of the well-known **says** operator.

In FSL, we enrich first-order logic with formulas of the kind

$$\{x\}\varphi(x) \textbf{ says } \psi \tag{1}$$

where $\{x\}\varphi(x)$ is a set-binding operator which represents the group composed by all the principals that satisfy $\varphi(x)$, **says** is a modal binary operator and ψ is a general formula.

Intuitively, we read Formula 1 as: "The group composed by all the principals that satisfy $\varphi(x)$ *supports* ψ". In this language is then possible to let principals jointly (as a group) support a statement, which is a desirable feature in several access control models as underlined in [5].

Previous approaches are limited in the representation of principals, in [17] principals are propositional atoms distributed in a lattice-based structure which can be combined with classical meet and join operators, in [7,14] a formula can be supported by at most one principal and it is not possible to make a group of principals jointly support a formula. In [18] groups of principals can be described by propositional atoms but their employment is limited to static and dynamic thresholds.

The proposed view on access control logics offers a general methodology to define policies and freedom in crafting logics. In fact we can let $\varphi(x)$ and ψ belong to two different languages $\mathbb{L}_p$ and $\mathbb{L}_e$ as language of principals and security expressions respectively which refer to two different systems (semantics). For instance we can think of formulas in $\mathbb{L}_p$ be SQL queries and formulas in $\mathbb{L}_e$ be Delegation Logic [18] expressions.

In order to formally specify how to evaluate expressions like 1, we formalize the **says** modality by using the fibring methodology [11] which, depending on the chosen languages (and systems), provides a formal tool to combine logics in a common framework which is coherent and does not collapse.

In this paper, in order to show the full expressiveness of our approach, we decide to make $\mathbb{L}_p = \mathbb{L}_e = \mathbb{L}$, where $\mathbb{L}$ is a classical first-order language plus the **says** operator. This approach offers us to nest the **says** modality and to express complex formulas in which free variables are shared between different levels of nesting of the **says**. So with $\mathbb{L}_p = \mathbb{L}_e$, in Formula 1, $\varphi(x)$ and ψ can share variables and φ may as well include occurrences of the **says** operator.

More formally, FSL formulae are defined by the following grammar

$$\varphi ::= \psi \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid \forall x(\varphi) \mid \{x\}\varphi(x) \textbf{ says } \varphi$$

where

- ψ is an ordinary first-order formula[5].
- $\varphi(x)$ denotes an FSL formula with one free variable x.

We now briefly show how we can employ FSL to identify groups of principals. The formula $\{x\}\varphi(x)$ is used to select the set of principals making the assertion **says**, to select a single principal whose name is A we write: $\{x\}(x = A)$ **says** s

The following formula means that all the *users* jointly (as a group) ask to delete $file_1$:

$$\{x\}user(x) \textbf{ says } delete(file_1)$$

Since $\varphi(x)$ and ψ can share variables, we can put restrictions on the variables occurring in ψ. E.g., the set of all users who all own file(s) y asks to delete the file(s) y:

$$\alpha(y) = \{x\}(user(x) \wedge own(x, y)) \textbf{ says } delete(y)$$

However, the formula above is satisfactory only in the particular situation where we are talking about the set of all users who assert **says** at once as a group (committee).

We can as well express that each member of a set identified by a formula can assert **says** separately. E.g., each user deletes individually the files he owns:

$$\forall x(user(x) \wedge own(x, y)) \rightarrow \{z\}(z = x) \textbf{ says } delete(y)$$

Note that the latter formula usually implies the former but not vice versa[6]. More generally, in FSL you have the possibility to express how the set $\{x|\varphi(x)holds\}$ says what it says. For instance, suppose we have $\varphi(x) = (x = A_1) \vee (x = A_2) \vee (x = A_3)$ then if at least one of $\{A_i\}$ **says** ψ is enough for the group to support ψ we add:

$$\{x\}\varphi(x) \textbf{ says } \psi \leftrightarrow \bigvee_{1 \leq i \leq 3} \{x\}(x = A_i) \textbf{ says } \psi$$

This represents the fact that each principal in the group can speak for the whole group.

In FSL, with features like the sharing of variables between $\varphi(x)$ and ψ, the nesting of the says and the employment of negation, we can express complex policies like separation of duties in a compact way. For instance, we can express the following: "A member m of the Program Committee can not accept a paper P_1 in which one of its authors says that he has published a paper with him after 2007":

$$\neg(\{m\}[PC(m) \wedge \{y\}author\_of(y, P_1) \textbf{ says } \exists p(paper(p) \wedge author\_of(m, p) \wedge author\_of(y, p) \wedge year(p) \geq 2007)] \textbf{ says } accept(P_1))$$

Due to space constraints, we leave to another paper the full treatment of the expressive power of the language by introducing abstractions to write access control policies (e.g., roles, delegation depth, thresholds).

[5] For a clear description of first-order languages we refer to [10].

[6] For instance, a committee may approve a paper that not all of its members would have accepted.

3 Modality Axioms

Despite the pervasive employment of the **says** operator in several logics for access control, only recently in [2] different axiomatizations of the says have been studied and proposed. The objective is to explore the formal consequences and the security interpretations of several possible axiomatizations, and thus to help in identifying access control logics that are sufficiently strong but not inconsistent, degenerate, or otherwise unreasonable.

Following we present the relevant axioms underlined in [2], notice that some of them are second-order due to a quantification over formulas. In Section 5 we will show how, by exploiting the multimodal approach of FSL, we can translate them as first-order constraints on the kripke semantics. We write $\Box_A X$ as an abbreviation for $\{x\}(x = A)$ **says** X, where X ranges over formulas.

(i) B *speaks for* A (notation $B \Rightarrow A$):

$$\forall X[\Box_B X \rightarrow \Box_A X].$$

This is the fundamental relation among principals in access control logics. If $B \Rightarrow A$, then all statements supported by B are also supported by A. This relation serves to form chains of responsibility: a program may speak for a user, much like a key may speak for its owner, much like a channel may speak for its remote end-point.

(ii) A *controls* ψ, for some specified formula ψ:

$$\Box_A \psi \rightarrow \psi$$

Intuitively it represents the direct control of A over a resource X. In this view it is desirable not to have a principal which controls *all* formulas, that is why we do not employ the universal quantifier.

(iii) *Hand-off axiom*:

$$\Box_A(B \Rightarrow A) \rightarrow (B \Rightarrow A)$$

Hand-off states that whenever A says that B speaks for A, then B does indeed speaks for A. This axiom allows every principal to decide which principals speak on its behalf, since it controls the delegation to other principals.

Sometimes this axiom follows from logic rules as in [2], sometimes it is assumed as an axiom. Note that the general axiom is too powerful, and thus risky for security: for example when A represents a group: if A **controls** $(B \Rightarrow A)$ then any member of A can add members to A. Thus, for instance, [3] does not adopt this axiom.

(iv) *Unit*:

$$\forall X[X \rightarrow \Box_A X]$$

Unit is stronger than the necessitation rule. In classical logic (but not intuitionistic), adopting Unit implies that each principal either always says the truth or it says the false: $(A \rightarrow B) \vee (B \rightarrow A)$. In the first case A speaks for any other principal, in the latter any other speaks for A. The policies described by this kind of systems are too manicheist.

(v) *Escalation*:

$$\forall X, Y[\Box_A X \rightarrow X \vee \Box_A Y]$$

Escalation is not considered as a desirable property. It amounts to "if A **says** ψ then ψ *or* A **says** *anything*": from A **says** ψ may follow a statement "much falser" than ψ. As an example of its riskiness (see [2]), consider that from (A **controls** ψ)$\wedge$(B **controls** ψ) we are allowed to infer that if A **says** B **says** ψ then ψ follows. If the logic is not able to avoid escalation, the only cumbersome solution is to make A avoid saying that B says ψ unless he really wishes to say ψ. Thus we must be careful that it does not follow from other properties (like from Unit or Bind in classical logics).

According to [2] in classical logic, Unit implies Escalation. If we leave out Unit we can rely on intermediate systems between the basic modal logic and Escalation. For instance, one may require the standard axiom C4 from modal logic ($\Box_A \Box_A X \rightarrow \Box_A X$) without obtaining Escalation. However, these intermediate systems appear quite limited in their support of delegation and related concepts.

In trying to have an expressive logic without Escalation as a theorem, intuitionism seems to be the right semantics to employ. In fact, Abadi in [2] propose a logic (CDD) based on second-order intuitionistic semantics in order to have a sufficiently expressive logic without having dangerous consequences like Escalation. In Section 4.2 we present predicate FSL which extends CDD expressiveness without using a second-order semantics.

4 The basic system FSL

This section introduces our basic system FSL step by step from a semantical point of view. First, in Section 4.1 we introduce modalities indexed by propositional atoms, then we take into account classical and intuitionistic models for the propositional setting and finally, in Section 4.2, we give a fibred semantics for modalities indexed by first-order formulas.

The FSL system can be defined with any logic $\mathbb{L}$ as a *Fibred Security System based on* $\mathbb{L}$. We will motivate the language for the cases of $\mathbb{L}$ = classical logic and $\mathbb{L}$ = intuitionistic logic. Basically adding the **says** connective to a system is like adding many modalities. So to explain and motivate FSL technically we need to begin with examining options for adding modalities to $\mathbb{L}$.

4.1 Adding modalities

We start by adding modalities to classical propositional logic. Let S be a nonempty set of possible worlds, for every subset $U \subseteq S$ consider a binary relation $R_U \subseteq S \times S$.

This defines a multimodal logic, containing at most 2^S modalities $\Box_U, U \subseteq S$. The models are of the form $(S, R_U, t_0, h), U \subseteq S$. In this view, if $U = \{t | t \vDash \varphi_U\}$ for some φ_U we get a modal logic with modalities indexed by formulas of itself. This requires now a formal definition.

Definition 4.1 [Language] Consider (classical or intuitionistic) propositional logic with the connectives $\wedge, \vee, \rightarrow, \neg$ and a binary connective $\Box_\varphi \psi$, where φ and ψ are

formulas.[7] The usual definition of wff is adopted.

Definition 4.2 We define classical Kripke models for this language.

(i) A model has the form $\mathbf{m} = (S, R_U, t_0, h), U \subseteq S$ where for each $U \subseteq S, R_U$ is a binary relation on S. $t_0 \in S$ is the actual world and h is an assignment, giving for each atomic q a subset $h(q) \subseteq S$.

(ii) We can extend h to all formulas by structural induction:
- $h(q)$ is already defined, for q atomic
- $h(A \wedge B) = h(A) \cap h(B)$
- $h(\neg A) = S - h(A)$
- $h(A \rightarrow B) = (S - h(A)) \cup h(B)$
- $h(A \vee B) = h(A) \cup h(B)$
- $h(\Box_\varphi \psi) = \{t|$ for all s $(tR_{h(\varphi)}s \rightarrow s \in h(\psi))\}$

(iii) $\mathbf{m} \vDash A$ iff $t_0 \in h(A)$.

Let us now do the same for intuitionistic logic. Here it becomes more interesting. An intuitionistic Kripke model has the form $\mathbf{m} = (S, \leq, t_0, h)$, where $(S, \leq)$ is a partially ordered set, $t_0 \in S$ and h is an assignment to the atoms such that $h(q) \subseteq S$. We require that $h(q)$ is a closed set, namely $x \in h(q)$ and $x \leq y \Rightarrow y \in h(q)$. Let D be a set, we can add for each $U \subseteq D$ a binary relation R_U on S. This semantically defines an intuitionistic modality, $\Box_U$.

In intuitionistic models we require the following condition to hold for each formula A, i.e. we want $h(A)$ to be closed: $x \in h(A)$ and $x \leq y \Rightarrow y \in h(A)$

This condition holds for A atomic and propagates over the intuitionistic connectives $\wedge, \vee, \rightarrow, \neg, \perp$. To ensure that it propagates over $\Box_U$ as well, we need an additional condition on R_U. To see what this condition is supposed to be, assume $t \vDash \Box_U A$. This means that: $\forall y(tR_U y \Rightarrow y \vDash A)$

Let $t \leq s$. If $s \nvDash \Box_U A$, then for some z such that $sR_U z$ we have $z \nvDash A$. This situation will be impossible if we require:

$$t \leq s \wedge sR_U z \Rightarrow tR_U z \qquad (*)$$

Put differently, if we use the notation: $R'_U(x) = \{y|xR_U y\}$ then

$$x \leq x' \Rightarrow R'_U(x) \supseteq R'_U(x') \qquad (*)$$

We now want to concentrate on what happens if U is defined by a formula φ_U, i.e. $U = h(\varphi_U)$. This will work only if U is closed, formally: $t \in U \wedge t \leq s \Rightarrow s \in U$

So from now on, we talk about modalities associated with closed subsets of S. We can now define our language. This is the same as defined in Definition 4.1. We now define the semantics.

Definition 4.3 A model has the form $\mathbf{m} = (S, \leq, R_U, t_0, h), U \subseteq S$ where $(S, \leq)$ is a partial order, $t_0 \in S$, and each $U \subseteq S$ is a closed set and so is $h(q)$ for atomic q. R_U satisfies condition (*) above. We define the notion $t \vDash A$ for a wff by induction, and then define $h(A) = \{t|t \vDash A\}$ So let's define $\vDash$:

[7] There are many such connectives, e.g. φ **says** $\psi, \varphi > \psi$ (conditional), $\bigcirc(\varphi/\psi)$ relative obligation, etc. The semantics given to it will determine its nature.

- $t \vDash q$ iff $t \in h(q)$
- $t \vDash A \wedge B$ iff $t \vDash A$ and $t \vDash B$
- $t \vDash A \vee B$ iff $t \vDash A$ or $t \vDash B$
- $t \vDash A \rightarrow B$ iff for all $s, t \leq s$ and $s \vDash A$ imply $s \vDash B$
- $t \vDash \neg A$ iff for all s, $t \leq s$ implies $s \nvDash A$
- $t \nvDash \bot$
- $t \vDash \Box_{\varphi}\psi$ iff for all s such that $tR_{h(\varphi)}s$ we have $s \vDash \psi$. We assume by induction that $h(\varphi)$ is known.
- $\mathbf{m} \vDash A$ iff $t_0 \vDash A$.

It is our intention to read $\Box_{\varphi}\psi$ as φ **says** ψ.

4.2 Predicate FSL

Intuitively, a predicate FSL fibred model is represented by a set of models linked together by means of a *fibring function*, every model has an associated domain D of elements together with a set of formulas that are true in it. In the FSL meta-model, the evaluation of the generic formula $\alpha = \{x\}\varphi(x)$ **says** ψ is carried out in two steps, first evaluating φ and then ψ in two different models. Suppose $\mathbf{m_1}$ is our (first-order) starting model in which we identify $U \subseteq D$ as the set of all the elements that satisfy φ. Once we have U we can access one or more worlds depending on the *fibring function* $\mathbf{f} : \mathcal{P}(D) \rightarrow \mathcal{P}(M)$ which goes from sets of elements in domain D to sets of models. At this point, for every model $\mathbf{m_i} \in \mathbf{f}(U)$ we must check that ψ is *true*, if this is the case then α is true in the meta-model.

The fact that in the same expression we evaluate different sub-formulas in different models is the core idea of the fibring methodology [11]. Think about a group of administrators that have to set up security policies for their company. From a semantical point of view, if we want to check if ψ holds in the depicted configuration by the administrators, we must

(i) Identify all the admins (all the elements that satisfy $admin(x)$).

(ii) Access the model that all the admins as a group have depicted.

(iii) Check in that model if ψ is *true* or *false*

Let $\mathbb{L}$ denote classical or intuitionistic predicate logic[8]. We assume the usual notions of variables, predicates, connectives $\wedge, \vee, \rightarrow, \neg$, quantifiers $\forall, \exists$ and the notions of free and bound variables.

Let $\mathbb{L}^+$ be $\mathbb{L}$ together with two special symbols:

- A binary (modality), **says**
- A set-binding operator $\{x\}\varphi(x)$ meaning the set of all x such that $\varphi(x)$

Definition 4.4 The language FSL has the following expressions:

(i) All formulas of $\mathbb{L}^+$ are level 0 formulas of FSL.

[8] Classical predicate logic and intuitionistic predicate logic have the same language. The difference is in the proof theory and in the semantics.

(ii) If $\varphi(x)$ and ψ are formulas of $\mathbb{L}^+$ then $\alpha = \{x\}\varphi(x)$ **says** ψ are level 1 'atomic' formulas of FSL. If $(x, x_1, \ldots, x_n)$ are free in φ and $y_1, \ldots, y_m$ are free in ψ then $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$ are free in α. The variable x in φ gets bound by $\{x\}$. The formula of level 1 are obtained by closure under the connectives and quantifiers of $\mathbb{L}^+$.

(iii) Let $\varphi(x)$ and ψ be formulas of FSL of levels r_1 and r_2 resp., then $\alpha = \{x\}\varphi$ **says** ψ is an 'atomic' formula of FSL of level $r = \max(r_1, r_2) + 1$.

(iv) Formulas of level n are closed under classical logic connectives and quantifiers of all 'atoms' of level $m \leq n$.

Definition 4.5 [FSL classical fibred model of level n]

(i) Any classical model with domain D is a FSL model of level 0.

(ii) Let $\mathbf{m}$ be a classical model of level 0 with domain D and let for each subset $U \subseteq D, \mathbf{f}^n(U)$ be a family of models of level n (with domain D). Then $(\mathbf{m}, \mathbf{f}^n)$ is a model of level $n + 1$.

Definition 4.6 [Classical satisfaction for FSL] We define satisfaction of formulas of level n in classical models of level $n' \geq n$ as follows.

First observe that any formula of level n is built up from atomic predicates of level 0 as well as 'atomic' formulas of the form $\alpha = \{x\}\varphi(x)$ **says** ψ, where φ and ψ are of lower level.

We therefore first have to say how we evaluate $(\mathbf{m}, \mathbf{f}^n) \vDash \alpha$. We assume by induction that we know how to check satisfaction in $\mathbf{m}$ of any $\varphi(x)$, which is of level $\leq n$. We can therefore identify the set $U = \{d \in D \mid \mathbf{m} \vDash \varphi(d)\}$.

Let $\mathbf{m}' \in \mathbf{f}^n(U)$. We can now evaluate $\mathbf{m}' \vDash \psi$, since ψ is of level $\leq n - 1$. So we say

$$(\mathbf{m}, \mathbf{f}^n) \vDash \alpha \text{ iff for all } \mathbf{m}' \in \mathbf{f}^n(U), \text{ we have } \mathbf{m}' \vDash \psi$$

We now define intuitionistic models for FSL. This will give semantics for the intuitionistic language.

Definition 4.7 We start with intuitionistic Kripke models which we assume for simplicity have a constant domain. The model $\mathbf{m}$ has the form $(S, \leq, t_0, h, D)$ where D is the domain and $(S, \leq, t_0)$ is a partial order with first point t_0 and h is an assignment function giving for each $t \in S$ and each m-place atomic predicate P a subset $h(t, P) \subseteq D^m$ such that $t_1 \leq t_2 \Rightarrow h(t_1, P) \subseteq h(t_2, P)$

We let $h(P)$ denote the function $\lambda t\ h(t, P)$. For $t \in S$ let

$$S_t = \{s \mid t \leq s\}$$
$$h(t, P) = h(P) \restriction S_t$$
$$\leq_t = \leq \restriction S_t$$

Where $\restriction$ represents the standard domain restriction.

Let $\mathbf{m}_t = (S_t, \leq_t, t, h_t, D)$. Note that a formula φ holds at $\mathbf{m} = (S, \leq, t_0, h, D)$ iff $t_0 \vDash \varphi$ according to the usual Kripke model definition of satisfaction.

(i) A model of level 0 is any model $\mathbf{m}$: $\mathbf{m} = (S, \leq, t_0, h, D)$.

(ii) Suppose we have defined the notion of models of level $m \leq n$, (based on the domain D).

We now define the notion of a model of level $n+1$

Let $\mathbf{m}$ be a model of level 0 with domain D. We need to consider not only $\mathbf{m}$ but also all the models $\mathbf{m}_t = (S_t, \leq_t, t, h_t, D)$, for $t \in S$. The definitions will be given simultaneously for all of them.

By an intuitionistic 'subset' of D in $(S, \leq, t_0, h, D)$, we mean a function $\mathbf{d}$ giving for each $t \in S$, a subset $\mathbf{d}(t) \subseteq D$ such that $t_1 \leq t_2 \Rightarrow \mathbf{d}(t_1) \subseteq \mathbf{d}(t_2)$.

Let $\mathbf{f}_t^n$ be a function associating with each $\mathbf{d}_t$ and $t \in S$ a family $\mathbf{f}_t^n(\mathbf{d}_t)$ of level n models, such that $t_1 \leq t_2 \Rightarrow \mathbf{f}_{t_1}^n(\mathbf{d}_{t_1}) \supseteq \mathbf{f}_{t_2}^n(\mathbf{d}_{t_2})$. Then $(\mathbf{m}_t, \mathbf{f}_t)$ is a model of level $n+1$ where $\mathbf{d}_t = \mathbf{d} \restriction S_t$.

Definition 4.8 [Satisfaction in fibred intuitionistic models] We define satisfaction of formulas of level n in models of level $n' \geq n$ as follows.

Let $(\mathbf{m}_t, \mathbf{f}_t^n)$ be a level n model. Let $\alpha = \{x\}\varphi(x)$ **says** ψ is of level n. We assume we know how to check satisfaction of $\varphi(x)$ in any of these models.

We can assume that

$$\mathbf{d}_t = \{x \in D \mid t \vDash \varphi(x) \text{ in } (\mathbf{m}_t, \mathbf{f}_t^n)\}$$

is defined. Then $t \vDash \alpha$ iff for all models $\mathbf{m}'_t$ in $\mathbf{f}_t^n(\mathbf{d}_t)$ we have $\mathbf{m}'_t \vDash \psi$.

5 Kripke Models for Axioms

In this section we show one advantage in employing the fibred semantics, we translate the most important second-order axioms appeared in [2] into first-order constraints on the kripke models. This result shows how, thanks to the multimodal semantics, second-order is not needed in dealing with axioms like *speaks-for* or *hand-off*, this could be useful in developing calculi to do automated reasoning on expressive access control policies.

In particular, we identify a precise mathematical relationship between well-known security axioms and semantical properties of FSL. From the point of view of logic engineering, it is important to see this relationship, because it helps one to understand the axioms being studied and how they affect the models of the underlying logic. From a practical point of view, we show that important security properties can be embodied in a decidable fragment of (multimodal) first-order logic. Decidability is a desirable property if we want security practitioners exploit theorem provers for access control procedures.

Suppose to have two intuitionistic modalities $\Box_A$ and $\Box_B$ and their accessibility relations R_A and R_B. So our Kripke model has the form $(S, \leq, R_A, R_B, t_0, h)$. We know for $\mu = A$ or $\mu = B$ that we have in the model

$$t \leq s \wedge sR_\mu z \rightarrow tR_\mu z. \qquad (*)$$

What other conditions can we impose on $\Box_\mu$?

(i) *The condition for the axiom Unit* $X \rightarrow \Box_\mu X$ is

$$xR_\mu y \rightarrow x \leq y \qquad (*1)$$

(ii) *The condition for the axiom C4* $\Box_A \Box_A x \rightarrow \Box_A$ is

$$xR_A y \wedge yR_A z \rightarrow zR_A y$$

(iii) *The condition for the axiom speaks-for* $\Box_B X \rightarrow \Box_A X$

$$xR_A y \rightarrow xR_B y \qquad (*2)$$

(iv) Note that $\Box_B X \rightarrow \Box_A X$ is taken in (*2) as an axiom schema. If we want to have $t \vDash \forall X(\Box_B X \rightarrow \Box_A X)$ i.e. we want $\Box_B \varphi \rightarrow \Box_A \varphi$ to hold at the point $t \in S$ for all wff φ, we need to require (*2) to hold above t, i.e.

$$\forall x, y(t \leq x \wedge xR_A y \rightarrow xR_B y) \qquad (*3)_t$$

(v) Consider now an axiom called *hand-off A to B*.

$$\Box_A(\forall X(\Box_B X \rightarrow \Box_A X)) \rightarrow \forall X(\Box_B X \rightarrow \Box_A X)$$

This axiom has a second order propositional quantifier in it.

The antecedent of the axiom wants $\Box_A(\forall X \Box_B X \rightarrow \Box_A X))$ to hold at t_0. This means in view of (3) above that $(*4_a)$ needs to hold

$$\forall t(t_0 R_A t \rightarrow (*3)_t) \qquad (*4_a)$$

The axiom says that if the antecedent holds at t_0 so does the consequent, i.e.

$$t_0 \vDash \forall X(\Box_B X \rightarrow \Box_A X).$$

We know the condition for that to hold is $(*3)_{t_0}$. Thus the condition for Hand-off A to B is

$$\forall t[t_0 R_A t \rightarrow (*3)_t] \rightarrow (*3)_{t_0} \qquad (*4)$$

The important point to note is that although the axiom is second order (has $\forall X$ in it both in the antecedent and consequence), the condition on the model is first order [9].

(vi) Concerning *Escalation* $\Box_A X \rightarrow X \vee \Box_A \bot$ its condition is

$$\exists y(xR_A y) \rightarrow xR_A x \qquad (*5)$$

To check whether we can have hand-off from A to B without escalation for A, for some choice of R_A and R_B, we need to check whether we can have (*4) without having (*5), for some wise choice of R_A and R_B. Thanks to a multi-modal semantics, we can then translate the second-order axioms in [2] into constraints on kripke models, this result show how first-order logic suffices to axiomatize the **says** modality in access control logics.

[9] Notice that we use first-order but we get a language more expressive than CDD[2] which is second-order.

6 Conclusion and Future Work

We have presented a logical formalism called FSL based on multimodal first-order logic. The proposed framework extends the expressivity of existing logics for access control by introducing sets of principals (described by formulas) as first-class citizen that can jointly support statements, and by permitting complex nesting of the **says** modality (see Section 2). FSL is based on a general methodology (i.e. fibring) to combine logics and use them within a unifying language [11], thanks to this approach we offer a general logical language to embody and extends existing access control formalisms. Thanks to the proposed semantics based on multimodalties, we showed that second-order logics are not necessary to model common axioms for the **says** like 'speaks-for' or 'hand-off'.

For instance, in [2] the presented calculus for the proposed (second-order) access control logic can not be employed in practical theorem proving due to its complexity. On the contrary, there are works in which first-order languages are constrainted in order to get nice computational results in the derivation time [15,18].

We studied how security axioms can be translated into first-order constraints on kripke models by introducing a model-driven study of logics for access control as underlined in [12].

As ongoing work we are formalizing the extension of well known logics like DL [18], SecPAL [7], DEBAC [8] and DKAL [14] with the FSL methodology to translate them into predicate FSL. In this view, FSL can be studied as a general framework to compare and integrate different logics for access control.

We are also working on providing a calculus for a tractable fragment of predicate FSL and on using FSL as a general language able to describe the many different instances of access control models by merging them with the meta-model proposed in [5].

References

[1] Abadi, M., *Access control in a core calculus of dependency*, Electr. Notes Theor. Comput. Sci. **172** (2007), pp. 5–31.

[2] Abadi, M., *Variations in access control logic*, in: R. van der Meyden and L. van der Torre, editors, *DEON*, LNCS **5076** (2008), pp. 96–109.

[3] Abadi, M., M. Burrows, B. W. Lampson and G. D. Plotkin, *A calculus for access control in distributed systems*, in: *Adcances in Cryptology (CRYPTO)*, LNCS **576** (1991), pp. 1–23.

[4] Abadi, M. and T. Wobber, *A logical account of ngscb*, in: D. de Frutos-Escrig and M. Núñez, editors, *Formal Techniques for Networked and Distributed Systems (FORTE)*, LNCS **3235** (2004), pp. 1–12.

[5] Barker, S., *The next 700 access control models or a unifying meta-model?*, ACM Symposium on Access Control Models and Technologies SACMAT 09 (2009), pp. 187–196.

[6] Bauer, L., M. A. Schneider, E. W. Felten and A. W. Appel, *Access control on the web using proof-carrying authorization*, in: *DARPA Information Survivability Conference and Exposition (DISCEX)* (2003), pp. 117–119.

[7] Becker, M. Y., C. Fournet and A. D. Gordon, *Design and semantics of a decentralized authorization language*, in: *IEEE Computer Security Foundations Symposium (CSF)* (2007), pp. 3–15.

[8] Bertolissi, C., M. Fernández and S. Barker, *Dynamic event-based access control as term rewriting*, in: S. Barker and G.-J. Ahn, editors, *Data and Applications Security (DBSec)*, LNCS **4602** (2007), pp. 195–210.

[9] Dekker, M. A. C. and S. Etalle, *Audit-based access control for electronic health records*, Electr. Notes Theor. Comput. Sci. **168** (2007), pp. 221–236.

[10] Enderton, H. B., "A Mathematical Introduction to Logic, 2nd Edition," Academic Press, 2000.

[11] Gabbay, D. M., "Fibring Logics," Oxford University Press, 1999.

[12] Garg, D. and M. Abadi, *A modal deconstruction of access control logics*, in: *Foundations of Software Science and Computational Structures (FoSSaCS)*, LNCS **4962** (2008), pp. 216–230.

[13] Garg, D., L. Bauer, K. D. Bowers, F. Pfenning and M. K. Reiter, *A linear logic of authorization and knowledge*, in: *European Symposium on Research in Computer Security (ESORICS)*, LNCS **4189** (2006), pp. 297–312.

[14] Gurevich, Y. and I. Neeman, *Dkal: Distributed-knowledge authorization language*, in: *IEEE Computer Security Foundations Symposium (CSF)* (2008), pp. 149–162.

[15] Halpern, J. Y. and V. Weissman, *Using first-order logic to reason about policies*, ACM Trans. Inf. Syst. Secur. **11** (2008).

[16] Lampson, B. W., *Computer security in the real world*, IEEE Computer **37** (2004), pp. 37–46.

[17] Lampson, B. W., M. Abadi, M. Burrows and E. Wobber, *Authentication in distributed systems: Theory and practice*, ACM Trans. Comput. Syst. **10** (1992), pp. 265–310.

[18] Li, N., B. N. Grosof and J. Feigenbaum, *Delegation logic: A logic-based approach to distributed authorization*, ACM Trans. Inf. Syst. Secur. **6** (2003), pp. 128–171.

[19] Wobber, E., M. Abadi and M. Burrows, *Authentication in the taos operating system*, ACM Trans. Comput. Syst. **12** (1994), pp. 3–32.