

Logic-Based Authorization (Thesis Proposal)

Deepak Garg

Committee:

Frank Pfenning (Chair),
Lujó Bauer, Anupam Datta,
Robert Harper, Martín Abadi

Computer Science Department
Carnegie Mellon University

June 06, 2008

Outline

1 Background

2 Logic design

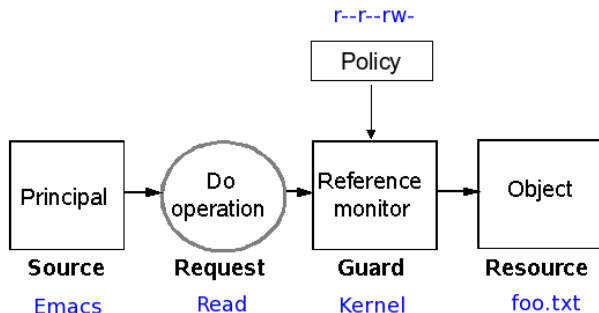
- Emphasis on proof-theory
- Intuitionistic logic vs classical logic
- The modality K says A
- Explicit time
- Linearity

3 Design of the file system

4 Outline of policy analysis

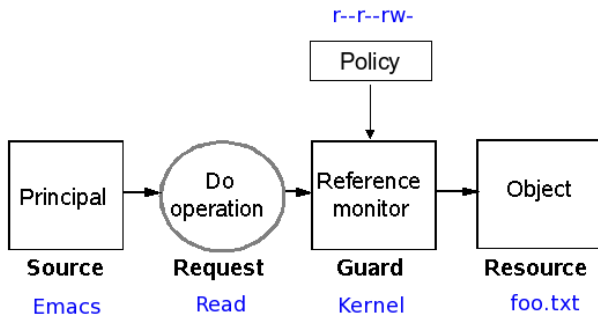
5 Summary of proposed work

The Problem of Access Control



- **Principal**
Abstraction for any entity (machine, user, program) that accesses resources or makes policies
- **Authorization Policy**
Rules/facts used to determine whether to allow a principal access to a resource

The Problem with Access Control



- Access control is pervasive in practice, hard to get right
 - ▶ Policy misinterpretations
 - ▶ Policy misconfigurations
- Basic idea: Use logic to represent policies [Lampson *et al.* '92]

Why Logic?

- Logic is **rigorous**
 - v. Textual descriptions of policies (ambiguous)
- Logic is **abstract**
 - v. Bit-level representations of policies (low-level, confusing)
- Logic is **flexible** and **extensible**
 - General connectives, may be combined
 - Reasoning constructs: linearity, constraints, ...
- Generic **enforcement** mechanism
 - Proofs!
- Amenable to **analysis**
 - Proof-theory (esp. sequent calculi, focusing)

Which Logic?

- First-order/propositional logic suffices (with exceptions)
 - ▶ Higher-order logic used sometimes, not necessary
- Authorization Logics
 - Specialized logics for representing authorization policies
- Some constructs have been proposed
- Some others in this thesis
- Eventual goal: capture as many policy motifs as possible in the logic

Representing Decentralized Policies

- Policies of different principals may interact to allow access
- Example
 - ① “Any I.P. address from CMU may download from ACM’s web portal”
 - ② “128.2.158.6 is a CMU I.P. address”
- Policy (1) made by ACM; policy (2) made by CMU
- How do we represent the difference?

- K says A (principal K states that formula A is true)
[Lampson *et al.* '92]
- Example
 - ① ACM says $\forall i. ((\text{CMU says}_{\text{myIP}(i)} \supset \text{maydownload}(i))$
 - ② CMU says $\text{myIP}(128.2.158.6)$

Enforcement: Proof-Carrying Authorization (PCA)

- Allow access only if there is a **formal proof** that authorization follows [Appel, Felten '99]

- Example

$$\Gamma = \text{ACM says } \forall i. ((\text{CMU says}_{\text{myIP}(i)}) \supset \text{maydownload}(i)) \\ \text{CMU says}_{\text{myIP}(128.2.158.6)}$$

- Allow access only if there is M such that

$$\Gamma \vdash M : \text{ACM says}_{\text{myIP}(128.2.158.6)}$$

- 128.2.158.6 **constructs** M ; gives to ACM
- ACM **verifies** M ; allows or denies access.
- Γ established through digitally signed certificates, if needed

Motivations and Summary of Proposed Work

- Much work on representing policies, ignoring the logic
 - Detailed study of a new logic, esp. proof-theory
- Unexplored policy motifs
 - Consumable credentials [GBBPR '06]
 - ▶ Time in policies [Deyoung,Garg,Pfenning '08]
- Analyze policies through using logic
 - Proof-theoretic methods
- Explore issues with PCA in a kernel
 - Prototype file system

Motivations and Summary of Proposed Work

- Much work on representing policies, ignoring the logic
 - Detailed study of a new logic, esp. proof-theory
- Unexplored policy motifs
 - Consumable credentials [GBBPR '06]
 - ▶ Time in policies [Deyoung,Garg,Pfenning '08]
- Analyze policies through using logic
 - Proof-theoretic methods
- Explore issues with PCA in a kernel
 - Prototype file system

“Logic, grounded in sound proof-theory, can be used to build a unified framework for specifying, enforcing, and analyzing authorization policies.”

Outline

1 Background

2 Logic design

- Emphasis on proof-theory
- Intuitionistic logic vs classical logic
- The modality K says A
- Explicit time
- Linearity

3 Design of the file system

4 Outline of policy analysis

5 Summary of proposed work

Outline

1 Background

2 Logic design

- Emphasis on proof-theory
- Intuitionistic logic vs classical logic
- The modality K says A
- Explicit time
- Linearity

3 Design of the file system

4 Outline of policy analysis

5 Summary of proposed work

Emphasis on Proof-Theory

- No clear metric to evaluate authorization logics
 - Lack of formal semantic model for policies
 - Only measure is “fitness” in application
- Proof-theory (esp. sequent calculus) and meta-theory (esp. cut-elimination) as basic criteria
- ▶ Assurance of the logic’s soundness (*à la* type-theory)
 - ▶ Proofs central to PCA
 - ▶ Useful in policy analysis
 - ▶ Proof-normalization useful in auditing proofs

Intuitionistic Logic vs Classical Logic

- Keep evidence as direct as possible
- Intuitionistic logic more appropriate
- Classical logic may hide evidence
 - ▶ Proofs by contradiction (A if $\neg\neg A$)
 - ▶ Prove $A \vee B$ without proving either A or B
 - ▶ Prove $\exists x.A$ without specifying x
- Intuitionistic logic better compatible with K says A
- Some policies fall in the Horn fragment

The Modality K says A

- What inference rules (or axioms) should we use?
- Trade-off between allowing
 - ▶ Too little (useless), e.g., treat K says A syntactically
 - ▶ Too much (dangerous), e.g., assume $A \equiv (K \text{ says } A)$
- Must have
 - ▶
$$\frac{\vdash A}{\vdash K \text{ says } A}$$
 - ▶ $\vdash (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B))$
- Must not have
 - ▶ $\nVdash (K \text{ says } A) \supset A$
- Questionable, sometimes
 - ▶ $\vdash A \supset (K \text{ says } A)$
- Useful, but optional
 - ▶ $\vdash (K \text{ says } K \text{ says } A) \supset K \text{ says } A$
 - ▶ $\vdash (K \text{ says } A) \supset K' \text{ says } K \text{ says } A$

The Proposed Logic: DTL (Propositional Fragment)

- Follow Martin-Löf's approach; judgments vs formulas
- Categorical judgments: (A true) and (K claims A)
- (K says A) true and K claims A are equivalent
- Hypothetical judgments: $\Gamma \xrightarrow{K} A$ true
"Assuming all claims of K are true, A follows from Γ "
- Basic rules (sequent calculus):

$$\frac{}{\Gamma, P \xrightarrow{K} P} \qquad \frac{\Gamma, K \text{ claims } A, A \xrightarrow{K} C}{\Gamma, K \text{ claims } A \xrightarrow{K} C}$$

$$\frac{\Gamma|_K \xrightarrow{K} A}{\Gamma \xrightarrow{K'} K \text{ says } A} \qquad \frac{\Gamma, K \text{ says } A, K \text{ claims } A \xrightarrow{K'} C}{\Gamma, K \text{ says } A \xrightarrow{K'} C}$$

Properties and Meta-Theory

- Natural deduction, proof-terms, local soundness and completeness
- Axiomatic system, Kripke semantics
- Admissibility of cut:
 - ▶ $\Gamma \xrightarrow{K} A$ and $\Gamma, A \xrightarrow{K} C$ imply $\Gamma \xrightarrow{K} C$
- Identity:
 - ▶ $\Gamma, A \xrightarrow{K} A$
- Subformula property
- Translations from many authorization logics to DTL

The Need for Explicit Time

- Policies often valid in specific intervals of time
- Examples:
 - 1 Alice is registered in 15212 from 01-16-08 to 05-07-08
 - 2 Students may access the registration web site from 6 AM to 9 PM
 - 3 Alice may enter Wean Hall from 9 AM to 6 PM on weekdays
- How is time-dependence represented?
- Option 1: Handle time outside the logic
 - ▶ Logically valid proofs may fail
 - ▶ Logic-based policy analysis ignores time
- Option 2: Internalize time into the logic
 - ▶ Use ideas from hybrid logic [Deyoung,Garg,Pfenning '08]

Representing Explicit Time

- Introduce first-order terms for intervals of time, I
- Refine judgments with time
 - ▶ $A \text{ true} \longrightarrow A \text{ on } I$
 - ▶ $K \text{ claims } A \longrightarrow K \text{ claims } A \text{ on } I$
- Partial order on intervals (subset order)
 - ▶ Formally modeled as constraints
- $\Psi; \Gamma \xrightarrow{K} A \text{ on } I$
- Modified init rule:
$$\frac{\Psi \models I \supseteq I'}{\Psi; \Gamma, P \text{ on } I \xrightarrow{K} P \text{ on } I'}$$
- Internalize $A \text{ on } I$ as $(A @ I) \text{ true}$

Consumable Credentials and Linearity

- Facts/policies that can be used once only
 - ▶ Alice may view FlickA on movie.com **once**.
 - ▶ Alice may enter my office door just this **once**.
 - ▶ Alice may act in the role of **either** faculty or administrator, but not both.
- How do we represent such policies?
- Use linear (or affine) logic [GBBPR '06]
- $\Gamma; \Delta \xrightarrow{K} A$
 - ▶ Γ : policies that may be used many times
 - ▶ Δ : policies that may be used once only
- Linearity ensures single use in *each proof*
- What about single use *across proofs*?
 - ▶ Difficult in distributed systems; no clear answer
 - ▶ Propose to use a centralized database in file system

Outline

- 1 Background
- 2 Logic design
 - Emphasis on proof-theory
 - Intuitionistic logic vs classical logic
 - The modality K says A
 - Explicit time
 - Linearity
- 3 Design of the file system
- 4 Outline of policy analysis
- 5 Summary of proposed work

File System with PCA

- Motivation:
 - ▶ Flexible access control is useful, necessary in some cases
 - ▶ Test bed for DTL (esp. linearity and time)
 - ▶ Explore issues in using PCA inside a kernel
- Retain System Call API (open, read, write, ...); make proofs optional
 - ▶ Attempt at full POSIX compliance
 - ▶ Existing programs can run unchanged
- Proofs are encoded in the file name
 - ▶ Must anticipate all uses when providing file name
- File system implemented with the Fuse toolkit
 - ▶ Proof checker: 700 lines of SML code + perl script
 - ▶ One parser, Fuse interface in C

Current Status and Proposed Work

- Current Status of Implementation
 - ▶ It works

Current Status and Proposed Work

- Current Status of Implementation

- ▶ It works
- ▶ ... slowly (approx 14ms per call without crypto)
- ▶ ... proofs have to be written by hand (in logical syntax)
- ▶ ... support for tracking linear credentials across calls incomplete
- ▶ ... logic is a fragment of DTL

Current Status and Proposed Work

- Current Status of Implementation

- ▶ It works
- ▶ ... slowly (approx 14ms per call without crypto)
- ▶ ... proofs have to be written by hand (in logical syntax)
- ▶ ... support for tracking linear credentials across calls incomplete
- ▶ ... logic is a fragment of DTL

- Proposed Work

- ▶ Benchmark, and try to improve efficiency
- ▶ Build a simple proof construction tool
- ▶ Implement a database for linear credentials

Outline

- 1 Background
- 2 Logic design
 - Emphasis on proof-theory
 - Intuitionistic logic vs classical logic
 - The modality K says A
 - Explicit time
 - Linearity
- 3 Design of the file system
- 4 **Outline of policy analysis**
- 5 Summary of proposed work

Logic-Based Policy Analysis

- Based on prior work, for a different logic [Garg,Pfenning '06]
- Develop meta-theorems to prove specific properties of policies
- Two-fold objective
 - ▶ Prove properties of policies
 - ▶ General criteria for evaluating authorization logics
- Example: Non-interference properties

$\Gamma, A \xrightarrow{K} C$ implies $\Gamma \xrightarrow{K} C$ if \langle some criteria on Γ, A, K, C \rangle

Example of a Non-interference Theorem

Suppose:

$\Gamma, K \text{ says } A \xrightarrow{K'} C$

There is no quantification on principals

$K \notin \Gamma, K', C$

Then, $\Gamma \xrightarrow{K'} C$

Example of a Non-interference Theorem

Suppose:

$\Gamma, K \text{ says } A \xrightarrow{K'} C$

There is no quantification on principals

$K \notin \Gamma, K', C$

Then, $\Gamma \xrightarrow{K'} C$

$\Gamma = \text{ACM says } \forall i. ((\text{CMU says } \text{myIP}(i)) \supset \text{maydownload}(i))$
 $\text{CMU says } \text{myIP}(128.2.158.6)$

Example of a Non-interference Theorem

Suppose:

$\Gamma, K \text{ says } A \xrightarrow{K'} C$

There is no quantification on principals

$K \notin \Gamma, K', C$

Then, $\Gamma \xrightarrow{K'} C$

$\Gamma = \text{ACM says } \forall i. ((\text{CMU says } \text{myIP}(i)) \supset \text{maydownload}(i))$
 $\text{CMU says } \text{myIP}(128.2.158.6)$

$\Gamma, \text{Alice says } A \xrightarrow{K} \text{ACM says } \text{maydownload}(X)$
implies
 $\Gamma \xrightarrow{K} \text{ACM says } \text{maydownload}(X)$

Outline

- 1 Background
- 2 Logic design
 - Emphasis on proof-theory
 - Intuitionistic logic vs classical logic
 - The modality K says A
 - Explicit time
 - Linearity
- 3 Design of the file system
- 4 Outline of policy analysis
- 5 Summary of proposed work

Summary of Proposed Work

- A new logic for writing authorization policies
 - ✓ Identify rules for K says A
 - ✓ Develop meta-theory and other properties
 - Versions with explicit time and linearity
- Implementation of a file system with PCA
 - ✓ Implement proof checker
 - ✓ Design interfaces
 - Benchmark, make efficient
 - Proof construction tool
 - Integrate linearity
- Policy analysis
 - Identify useful analyses
 - Develop meta-theorems and logical methods

Time estimate: 12 to 15 months

“Logic, grounded in sound proof-theory, can be used to build a unified framework for specifying, enforcing, and analyzing authorization policies.”

Questions?

Extra Slides

Intuitionistic Logic over Classical Logic

- Normal form of proofs gives useful information
 - ▶ No proofs by contradiction
 - ▶ $\vdash A \vee B$ implies $\vdash A$ or $\vdash B$

- Example

p_1 : $\forall i. (\text{student}(i) \supset \text{mayread}(i, \text{hiring_schedule}))$

p_2 : $\forall i. (\text{faculty}(i) \supset \text{mayread}(i, \text{hiring_schedule}))$

M = $\lambda x : (\text{student}(\text{Alice}) \vee \text{faculty}(\text{Alice})).$

case x of

inl y \Rightarrow p_1 y

| inr z \Rightarrow p_2 z

- ▶ Alice constructs $N : (\text{student}(\text{Alice}) \vee \text{faculty}(\text{Alice}))$
- ▶ Submits proof $(M N)$

Intuitionistic Logic over Classical Logic

- Normal form of proofs gives useful information

- ▶ No proofs by contradiction
- ▶ $\vdash A \vee B$ implies $\vdash A$ or $\vdash B$

- Example

p_1 : $\forall i. (\text{student}(i) \supset \text{mayread}(i, \text{hiring_schedule}))$

p_2 : $\forall i. (\text{faculty}(i) \supset \text{mayread}(i, \text{hiring_schedule}))$

M = $\lambda x : (\text{student}(\text{Alice}) \vee \text{faculty}(\text{Alice})).$

case x of

inl y \Rightarrow p_1 y

| inr z \Rightarrow p_2 z

- ▶ Alice constructs $N : (\text{student}(\text{Alice}) \vee \text{faculty}(\text{Alice}))$
- ▶ Submits proof $(M N)$
- ▶ N normalizes to $(\text{inl } N_1)$ or $(\text{inr } N_2)$
- ▶ $(M N)$ normalizes to $(p_1 N_1)$ or $(p_2 N_2)$

Intuitionistic Logic over Classical Logic

- Normal form of proofs gives useful information
 - ▶ No proofs by contradiction
 - ▶ $\vdash A \vee B$ implies $\vdash A$ or $\vdash B$

- Example

p_1 : $\forall i. (\text{student}(i) \supset \text{mayread}(i, \text{hiring_schedule}))$

p_2 : $\forall i. (\text{faculty}(i) \supset \text{mayread}(i, \text{hiring_schedule}))$

M = $\lambda x : (\text{student}(\text{Alice}) \vee \text{faculty}(\text{Alice})).$

case x of

inl y \Rightarrow p_1 y

| inr z \Rightarrow p_2 z

- ▶ Alice constructs $N : (\text{student}(\text{Alice}) \vee \text{faculty}(\text{Alice}))$
- ▶ Submits proof $(M N)$
- ▶ N normalizes to $(\text{inl } N_1)$ or $(\text{inr } N_2)$
- ▶ $(M N)$ normalizes to $(p_1 N_1)$ or $(p_2 N_2)$

- Classical logic does not provide this guarantee

Inference Rules for Propositional DTL

$$\frac{P \text{ atomic}}{\Gamma, P \xrightarrow{K} P} \text{init}$$

$$\frac{\Gamma, K \text{ claims } A, A \xrightarrow{K'} C \quad K \succeq K'}{\Gamma, K \text{ claims } A \xrightarrow{K'} C} \text{claims}$$

$$\frac{\Gamma \mid_K \xrightarrow{K} A}{\Gamma \xrightarrow{K'} K \text{ says } A} \text{saysR}$$

$$\frac{\Gamma, K \text{ says } A, K \text{ claims } A \xrightarrow{K'} C}{\Gamma, K \text{ says } A \xrightarrow{K'} C} \text{saysL}$$

$$\frac{\Gamma \xrightarrow{K} A \quad \Gamma \xrightarrow{K} B}{\Gamma \xrightarrow{K} A \wedge B} \wedge R$$

$$\frac{\Gamma, A \wedge B, A, B \xrightarrow{K} C}{\Gamma, A \wedge B \xrightarrow{K} C} \wedge L$$

$$\frac{\Gamma \xrightarrow{K} A}{\Gamma \xrightarrow{K} A \vee B} \vee R_1$$

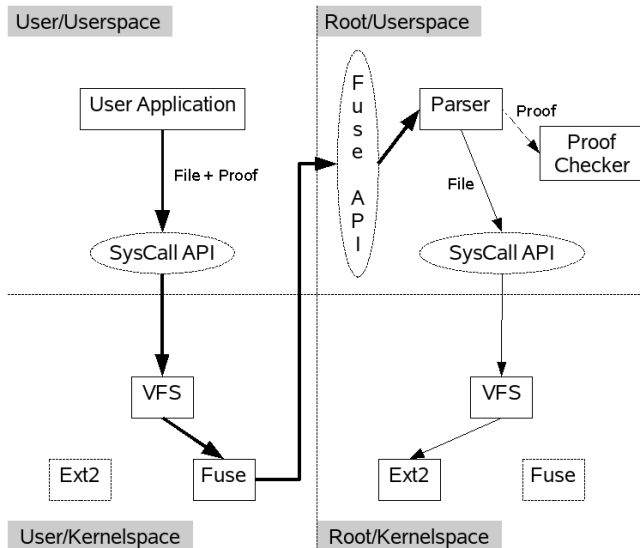
$$\frac{\Gamma \xrightarrow{K} B}{\Gamma \xrightarrow{K} A \vee B} \vee R_2$$

$$\frac{\Gamma, A \vee B, A \xrightarrow{K} C \quad \Gamma, A \vee B, B \xrightarrow{K} C}{\Gamma, A \vee B \xrightarrow{K} C} \vee L$$

$$\frac{\Gamma, A \xrightarrow{K} B}{\Gamma \xrightarrow{K} A \supset B} \supset R$$

$$\frac{\Gamma, A \supset B \xrightarrow{K} A \quad \Gamma, A \supset B, B \xrightarrow{K} C}{\Gamma, A \supset B \xrightarrow{K} C} \supset L$$

Architecture



Passing Proofs to FS calls

- No possibility of dialog; we want to retain the system call API
- Bob wants to read `foo.pdf`
- `proof_file` contains proofs of access to `foo.pdf`
- Encode proof file in the file name
- Syntax: `@:foo.pdf@proof_file:@foo.pdf`
- Multiple components:
`/usr0/ @:dg@pf1:@dg/ @:foo.pdf@pf2:@foo.pdf`
- Work with all calls, including system calls (open, read, write, ...) and shell commands
 - ▶ `@` and `:` are legitimate characters in file names, but do not usually occur in them

Proof files

- `/usr0/ @:dg@pf1:@dg/ @:foo.pdf@pf2:@foo.pdf`

- Sample file pf1:

```
%exec "/usr0/dg" [List of certificates] <Proof>
```

- Sample file pf2:

```
%read "/usr0/dg/foo.pdf" [List of certificates] <Proof>
```

```
%write "/usr0/dg/foo.pdf" [List of certificates] <Proof>
```

What do I prove?

- Proposition to be proved is uniquely determined by four things:
 - ▶ Caller's process id
 - ▶ File/directory accessed
 - ▶ Permission requested (read, write or execute)
 - ▶ Current time
- Example: to *read* file `foo.pdf` at time t , proc id 145 must show that:
 $\langle \text{admin}, [t, t] \rangle \text{may_read}(\text{int2pid}(145), \text{str2file}(\text{foo.pdf}))$
- No nonces!

Difficulties: Public Key Infrastructure (PKI)

- Why a PKI?
 - ▶ Certificates have to be signed and verified!
- Which PKI should we use?
 - ▶ PCA is independent of PKI
 - ▶ We use GnuPG (open source, no CA)
- How do we relate users to public keys?
 - ▶ Currently hardcoded
 - ▶ Practically from a database/file
- What uniquely identifies a principal – key or user id?
 - ▶ User id in our implementation (to make policies intuitive)

Difficulties: User Defined Predicates

- Bob wants to create a predicate $\text{friendBob}(K)$, meaning that K is a friend of Bob
- How does Bob specify this?
- How does Bob tell the file system how to verify the predicate?
- Can Alice make the statement $\text{friendBob}(\text{Charlie})$?