Principal-Centric Reasoning in Constructive Authorization Logic

Deepak Garg

Carnegie Mellon University

June 23, 2008
Outline

1. Background

2. Logic design: proof-theory
Outline

1. Background

2. Logic design: proof-theory
Policy is central – how do we represent it?
- Common methods: bit-level encodings, access control lists
- Low-level, confusing; may lead to programming and administrative errors

Represent policies as formulas in a logic
- Rigorous, high-level, flexible
- Can we use the logical representation for enforcement?
- What logic should be we use?
Enforcement through Proofs

- Allow access iff there is a formal proof \[\text{[Blaze et al. '96]}\]
- Let \( \Gamma \) denote the set of formulas representing policies
- Allow access iff \( \Gamma \vdash A \) (\( A \) depends on requester, object, and operation)

- Who constructs the proof?
  - Put burden of proof construction on requesting principal
    - Proof-carrying authorization \[\text{[Appel, Felten '99]}\]
  - Requester **constructs** and submits proof with request
  - Reference monitor **verifies** proof
    - Proof good \( \Rightarrow \) allow access
    - Proof bad \( \Rightarrow \) deny access

Proofs are a generic enforcement mechanism for access control
Which Logic?

- **First-order/propositional** vs. higher-order
- **Intuitionistic** vs. classical

- Does first-order intuitionistic logic suffice?
  - Almost, but some policy motifs need other constructs
Which Logic?

- First-order/propositional vs. higher-order
- Intuitionistic vs. classical

Does first-order intuitionistic logic suffice?

- Almost, but some policy motifs need other constructs
Which Logic?

- First-order/propositional vs. higher-order
- Intuitionistic vs. classical

Does first-order intuitionistic logic suffice?

- Almost, but some policy motifs need other constructs
Decentralization: The Need for Modalities

- Policies may be made by different principals
- Example (Can Alice visit USA? [without a visa])
  1. “Any Canadian national may visit USA without a visa”
  2. “Alice is a Canadian national”
- Policy (1) made by USA; policy (2) made by Canada
- How do we represent the difference?

- $K$ says $A$ (principal $K$ states that formula $A$ is true)
- Family of principal-indexed modalities [Lampson et al. ’92]
- Example
  1. USA says $\forall i. ((\text{Canada says myCitizen}(i)) \supset \text{mayVisit}(i))$
  2. Canada says myCitizen(Alice)

Which modal logic?
Decentralization: The Need for Modalities

- Policies may be made by different principals
- Example (Can Alice visit USA? [without a visa])
  1. “Any Canadian national may visit USA without a visa”
  2. “Alice is a Canadian national”
- Policy (1) made by USA; policy (2) made by Canada
- How do we represent the difference?

\( K \text{ says } A \) (principal \( K \) states that formula \( A \) is true)

- Family of principal-indexed modalities [Lampson et al. ’92]
- Example
  1. USA \text{ says } \forall i. ((\text{Canada says } \text{myCitizen}(i)) \supset \text{mayVisit}(i))
  2. Canada \text{ says } \text{myCitizen}(\text{Alice})

Which modal logic?
Decentralization: The Need for Modalities

- Policies may be made by different principals
- Example (Can Alice visit USA? [without a visa])
  1. “Any Canadian national may visit USA without a visa”
  2. “Alice is a Canadian national”
- Policy (1) made by USA; policy (2) made by Canada
- How do we represent the difference?

- $K$ says $A$ (principal $K$ states that formula $A$ is true)
- Family of principal-indexed modalities [Lampson et al. ’92]
- Example
  1. USA says $\forall i. ((\text{Canada says } \text{myCitizen}(i)) \supset \text{mayVisit}(i))$
  2. Canada says $\text{myCitizen}(\text{Alice})$

Which modal logic?
Choosing the “Right” $K$ says $A$

- Trade-off between
  - Weak modalities (useless), e.g., treat $K$ says $A$ syntactically
  - Strong modalities (dangerous), e.g., assume $A \equiv (K$ says $A)$

- Some past approaches:
  - □ from modal logic $K$ (classical) [Lampson et al. ’92, ’93]
    \[
    \vdash A \\
    \vdash K \text{ says } A \\
    \vdash (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B))
    \] (nec)
  - Original PCA (classical) [Appel, Felten ’99]
    \[
    \vdash A \supset (K \text{ says } A) \\
    \vdash (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B))
    \] (unit) (K)
  - ○ from lax logic (intuitionistic) [GF ’06, Abadi ’06, …]
    \[
    \vdash A \supset (K \text{ says } A) \\
    \vdash (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B))
    \] (unit) (K)
    \[
    \vdash (K \text{ says } K \text{ says } A) \supset (K \text{ says } A)
    \] (C4)
  - …

- No clear metric to evaluate “fitness” of modality
Choosing the “Right” $K$ says $A$

- Trade-off between
  - Weak modalities (useless), e.g., treat $K$ says $A$ syntactically
  - Strong modalities (dangerous), e.g., assume $A \equiv (K$ says $A$)

- Some past approaches:
  - □ from modal logic $K$ (classical) [Lampson et al. ’92, ’93]
    \[
    A \vdash K \text{ says } A
    \]
    \[
    (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B))
    \]
  - Original PCA (classical) [Appel, Felten ’99]
    \[
    A \supset (K \text{ says } A)
    \]
    \[
    (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B))
    \]
  - ○ from lax logic (intuitionistic) [GF ’06, Abadi ’06, …]
    \[
    A \supset (K \text{ says } A)
    \]
    \[
    (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B))
    \]
    \[
    (K \text{ says } K \text{ says } A) \supset (K \text{ says } A)
    \]
  - …

- No clear metric to evaluate “fitness” of modality
In this paper . . .

- A new propositional logic $\text{DTL}_0$ for representing access control policies
  - Sequent calculus and Hilbert-style axiomatization
  - Meta-theory: cut-elimination
  - Kripke semantics
  - Translations from other authorization logics to $\text{DTL}_0$
  - Closely related to constructive S4

- Emphasis on proof-theory esp. sequent calculus and cut-elimination
  - Proofs central to enforcement
  - Assurance of the logic’s soundness (à la Martin-Löf’s type-theory)
In this paper . . .

- A new propositional logic $\text{DTL}_0$ for representing access control policies
  - Sequent calculus and Hilbert-style axiomatization
  - Meta-theory: cut-elimination
  - Kripke semantics
  - Translations from other authorization logics to $\text{DTL}_0$
  - Closely related to constructive S4

- Emphasis on proof-theory esp. sequent calculus and cut-elimination
  - Proofs central to enforcement
  - Assurance of the logic’s soundness (à la Martin-Löf’s type-theory)
Outline

1. Background

2. Logic design: proof-theory
DTL$_0$: Syntax and Axiomatic System

\[ A, B ::= P \mid A \land B \mid A \lor B \mid \top \mid \bot \mid A \supset B \mid K \text{ says } A \]

Axiomatic System

\[ \vdash A \]
\[ \vdash K \text{ says } A \] (nec)
\[ \vdash (K \text{ says } (A \supset B)) \supset ((K \text{ says } A) \supset (K \text{ says } B)) \] (K)
\[ \vdash (K \text{ says } A) \supset (K \text{ says } K \text{ says } A) \] (4)
\[ \vdash K \text{ says } ((K \text{ says } A) \supset A) \] (C)

(C) stands for “conceit”

Replacing (C) with \((K \text{ says } A) \supset A\) gives CS4
Sequent Calculus

- Follow Martin-Löf’s method: separate formulas from judgments
  - Judgments are predicates over formulas, and evidenced by proofs

- Categorical (non-hypothetical) judgments:
  - $A \text{ true}$ (Usually elide the name true)
  - $K$ claims $A$ ($K$ claims $A \equiv (K$ says $A) \text{ true}$)
  - $K$ says $A$ internalizes $K$ claims $A$ into the syntax of formulas

- Hypothetical judgments or sequents: $\Gamma \xrightarrow{K} A \text{ true}$

  $\Gamma ::= \cdot \mid \Gamma, A \text{ true} \mid \Gamma, K \text{ claims } A$

- "If we assume that $K$ claims $A$ implies $A$ for each $A$, then assumptions $\Gamma$ entail $A$"
  - $K$ is called the context of the sequent
Basic Principles

- Context principle: In context $K$, $K$ claims $A$ entails $A$ true

- Claim principle: $K$ claims $A$ holds (in any context) if using only claims of $K$, we can prove $A$ true in context $K$.

- Internalization principle: $K$ claims $A$ is equivalent to ($K$ says $A$) true
Basic Rules

\[
\begin{align*}
\Gamma, P \text{ true} & \quad \Rightarrow \quad P \text{ true} \\
\Gamma, K \text{ claims } A, A \text{ true} & \quad \Rightarrow \quad C \text{ true} \\
\Gamma, K \text{ claims } A & \quad \Rightarrow \quad C \text{ true}
\end{align*}
\]

Context principle: In context \( K \), \( K \text{ claims } A \) entails \( A \text{ true} \)
Rules for implication

\[
\Gamma, A \text{ true } \xrightarrow{K} B \text{ true} \\
\Gamma \xrightarrow{K} (A \supset B) \text{ true} \\
\Gamma, (A \supset B) \text{ true } \xrightarrow{K} A \text{ true} \quad \Gamma, B \text{ true } \xrightarrow{K} C \text{ true} \\
\Gamma, (A \supset B) \text{ true } \xrightarrow{K} C \text{ true} \\
\]

- Rules for other propositional connectives as usual
Rules for says

\[ \Gamma |_K K \rightarrow A \text{ true} \]

\[ \frac{\Gamma \rightarrow (K \text{ says } A) \text{ true}}{\text{ saysR}} \]

where \( \Gamma |_K = \{K \text{ claims } C \mid (K \text{ claims } C) \in \Gamma\} \)

\[ \Gamma, (K \text{ says } A) \text{ true}, K \text{ claims } A \rightarrow C \text{ true} \]

\[ \frac{\Gamma, (K \text{ says } A) \text{ true} K' \rightarrow C \text{ true}}{\text{ saysL}} \]

Internalization principle: \( K \text{ claims } A \) is equivalent to \( (K \text{ says } A) \text{ true} \)

Claim principle: \( K \text{ claims } A \) holds (in any context) if using only claims of \( K \), we can prove \( A \text{ true} \) in context \( K \).
Summary of Rules

\[\Gamma, P \text{ true } \xrightarrow{K} P \text{ true}\]  \hspace{2cm} \[\Gamma, K \text{ claims } A, A \text{ true } \xrightarrow{K} C \text{ true}\]

\[\Gamma, A \text{ true } \xrightarrow{K} B \text{ true}\]  \hspace{2cm} \[\Gamma, (A \supset B) \text{ true } \xrightarrow{K} A \text{ true}\]

\[\Gamma \xrightarrow{K} (A \supset B) \text{ true}\]  \hspace{2cm} \[\Gamma, B \text{ true } \xrightarrow{K} C \text{ true}\]

\[\Gamma \xrightarrow{K} A \text{ true}\]  \hspace{2cm} \[\Gamma, (A \supset B) \text{ true } \xrightarrow{K} C \text{ true}\]

\[\Gamma, (K \text{ says } A) \text{ true, } K \text{ claims } A \xrightarrow{K'} C \text{ true}\]  \hspace{2cm} \[\Gamma, (K \text{ says } A) \text{ true } \xrightarrow{K'} C \text{ true}\]

\[\Gamma \xrightarrow{K'} (K \text{ says } A) \text{ true}\]  \hspace{2cm} \[\Gamma, (K \text{ says } A) \text{ true } \xrightarrow{K'} C \text{ true}\]

\[\Gamma, K \text{ claims } A \xrightarrow{K} C \text{ true}\]  \hspace{2cm} \[\Gamma, (K \text{ says } A) \text{ true } \xrightarrow{K'} C \text{ true}\]
Properties and Meta-Theory

- **Admissibility of cut:**
  - $\Gamma \vdash K A \text{ true}$ and $\Gamma, A \text{ true} \vdash K C \text{ true}$ imply $\Gamma \vdash K C \text{ true}$

- **Identity:**
  - $\Gamma, A \text{ true} \vdash K A \text{ true}$

- **Connection between axiomatic system and sequent calculus**
  - $\Gamma \vdash K A$ if and only if $\vdash K \text{ says } (\Gamma \supset A)$
  - Actually an *embedding theorem*

- **Sound and complete Kripke semantics in the paper**
  - Adapt Alechina *et al.*’s Kripke semantics for CS4
  - No ♦; need to associate principals with worlds

- **Sound and complete translations from other authorization logics**
DTL₀ Generalizes CS4 (Without ♦)

- In the special case of only one principal (say ℓ), DTL₀’s sequent calculus reduces to that for CS4.

\[
\begin{align*}
\Gamma, P \text{ true} & \xrightarrow{K} P \text{ true} & \Gamma, K \text{ claims } A, A \text{ true} & \xrightarrow{K} C \text{ true} \\
\Gamma, A \text{ true} & \xrightarrow{K} B \text{ true} & \Gamma, (A \supset B) \text{ true} & \xrightarrow{K} C \text{ true}
\end{align*}
\]

\[
\begin{align*}
\Gamma \xrightarrow{K} (A \supset B) \text{ true} & \quad \Gamma, B \text{ true} \xrightarrow{K} C \text{ true} \\
\Gamma |_K \xrightarrow{K} A \text{ true} & \quad \Gamma, (K \text{ says } A) \text{ true, } K \text{ claims } A \xrightarrow{K'} C \text{ true}
\end{align*}
\]

\[
\begin{align*}
\Gamma \xrightarrow{K} (K \text{ says } A) \text{ true} & \quad \Gamma, (K \text{ says } A) \text{ true} \xrightarrow{K'} C \text{ true}
\end{align*}
\]
DTL₀ Generalizes CS4 (Without ♦)

In the special case of only one principal (say ℓ), DTL₀’s sequent calculus reduces to that for CS4.

\[
\begin{align*}
\text{init} & : \Gamma, P \text{ true } \vdash P \text{ true} \\
\Gamma, A \text{ true } \vdash B \text{ true} & : \Gamma \vdash (A \supset B) \text{ true} \\
\Gamma|_{ℓ} \vdash A \text{ true} & : \Gamma \vdash (ℓ \text{ says } A) \text{ true} \\
\Gamma, ℓ \text{ claims } A, A \text{ true } \vdash C \text{ true} & : \Gamma, ℓ \text{ claims } A \vdash C \text{ true} \\
\Gamma, (A \supset B) \text{ true } \vdash A \text{ true} & : \Gamma, B \text{ true } \vdash C \text{ true} \\
\Gamma, (ℓ \text{ says } A) \text{ true } \vdash C \text{ true} & : \Gamma, (ℓ \text{ says } A) \text{ true } \vdash C \text{ true} \\
\end{align*}
\]
DTL$_0$ Generalizes CS4 (Without ♦)

- In the special case of only one principal (say ℓ), DTL$_0$’s sequent calculus reduces to that for CS4.

\[
\begin{align*}
\Gamma, P \text{ true} & \rightarrow P \text{ true} & \Gamma, A \text{ valid, } A \text{ true} & \rightarrow C \text{ true} \\
\Gamma \rightarrow (A \supset B) \text{ true} & \quad \Gamma, (A \supset B) \text{ true} & \rightarrow A \text{ true} & \quad \Gamma, B \text{ true} & \rightarrow C \text{ true} \\
\Gamma | & \rightarrow A \text{ true} & \Gamma, (\square A) \text{ true, } A \text{ valid} & \rightarrow C \text{ true} \\
\Gamma & \rightarrow (\square A) \text{ true} & \Gamma, (\square A) \text{ true} & \rightarrow C \text{ true}
\end{align*}
\]
DTL\textsubscript{0} \Rightarrow \text{CS4}^m

- The following embedding is sound and complete

\[\neg K \text{ says } A^\dashv = \Box_K (g_K \supset \neg A^\dashv)\]

- (CS4\textsuperscript{m} \Rightarrow DTL\textsubscript{0}) Unknown!
Summary and Future Work

- New logic for writing authorization policies
- Unusual, but expressive and (hopefully) useful
- Emphasis on proof-theory; sequent calculus, meta-theoretic properties
- Kripke semantics
- Connections to other logics

Part of a larger project
- More logical primitives: linearity, time
- Applications to real policies
- Implementation of a file system using the logic
- Using proof-theory to prove properties of policies
Thank You

Questions?
Rules without judgments

\[ \Gamma, P \xrightarrow{K} P \]

\[ \Gamma, A \xrightarrow{K} B \quad \xRightarrow{\text{init}} \quad \Gamma \xrightarrow{K} A \cup B \]

\[ \Gamma, A \cup B \xrightarrow{K} A \quad \text{R} \]

\[ \Gamma, B \xrightarrow{K} C \quad \text{L} \]

\[ K \text{ says } \Gamma \xrightarrow{K} A \quad \xRightarrow{\text{saysR}} \quad K \text{ says } \Gamma, \Gamma' \xrightarrow{K'} K \text{ says } A \]

\[ \Gamma, K \text{ says } A, A \xrightarrow{K} C \quad \xRightarrow{\text{saysL}} \quad \Gamma, K \text{ says } A \xrightarrow{K} C \]