Constructive Authorization Logics

Frank Pfenning

Carnegie Mellon University

Invited Talk
Workshop on Foundations of Computer Security (FCS'05)
Chicago, Illinois, June 30-July 1, 2005

Joint work with Deepak Garg and Kevin Watkins

Work in progress!
Outline

• Background
• Towards Universal Access Control
• Desiderata for Authorization
• Proof-Carrying Authorization
• Logic Design Principles
• Intuitionistic Authorization Logic
• Cut Elimination
• Independence and Non-Interference
• Most Closely Related Work
• Conclusion
Authentication and Authorization

- **Authentication**: who made a statement
  - Public key cryptography
  - Signed certificates

- **Authorization**: who should gain access to resource
  - Access control lists
  - Trust management
  - Relies on authentication
Authorization Logics

- **Authorization logics** provide a high-level, formal approach to access control in distributed systems

- Unifying basis for “EEE”
  - *Expressing* access control policy
  - *Enforcing* access control policy
  - *Exploring* consequences of access control policy

- Abstract away from
  - Mechanisms for authentication
  - Communication media and encryption
  - Protocols
Our Project

- Distributed System Security via Logical Frameworks
- PIs: Lujo Bauer, Mike Reiter, Frank Pfenning
- Supported by ONR N00014-04-1-0724 and NSF Cybertrust Center
- Using *smart phones* as “universal” access control device
  - Office door, computer (right now!)
Sample Scenario

- Office door lock equipped with Bluetooth device
- Principal with smart phone approaches door
- Mutual discovery protocol
- Authorization dialog
- Door opens (or not)
- Implemented on CyLab floor, CiC, CMU
Sample Access Control Policy

- I can access my office
- The department head can access my office
- My secretary can access my office
- I trust my secretary to let others into my office
- My students can access my office
- The floor marshal can access my office
- I trust my wife in all things
- Anyone may ask me to get into my office
Desiderata for Authorization

- Expression, Enforcement, Exploration (EEE)
  - Expressive policy language
  - Simple enforcement of policies
  - Feasible reasoning about policies

- Extensibility

- Small trusted computing base

- Smooth integration of authentication

- Work with distributed information
Proof-Carrying Authorization

- Proof-carrying authorization
  [Appel & Felten’99] [Bauer’03]
- Express policy in authorization logic
- Prove right to access resource within logic
- Transmit actual proof object to resource
- Check proof object to grant access
- Authentication via signed statements
- First demonstration with web browser
  [Bauer, Schneider, Felten’02]
Scenario Revisited

- WeH 8117 is Frank’s office
- WeH 8117 equipped with Bluetooth device
- Walk through two simple exchanges
- Illustrate basic ideas
- Ignoring discovery
- Ignoring freshness, nonces, etc.
- Handled in implementation
"I can open my office"

- Policy: *I can open my office*
- Frank approaches WeH 8117 with smart phone
- WeH 8117 challenges with
  
  \(? : \text{frank says open}(\text{frank}, \text{weh.8117})\)

- Policy embodied in challenge
- Frank signs
  
  \(\text{frank says open}(\text{frank}, \text{weh.8117})\)
  
  to obtain c38d9103294
"I can open my office"

- Frank replies
  \[x509(\text{c38d9103294})\]
- WeH 8117 checks (trivial) proof
  \[x509(\text{c38d9103294}) : \text{frank says open(frank, weh.8117)}\]
- Door opens
- Proof checking requires certificate checking for authentication
"My secretary can open my office"

- Policy: *My secretary can open my office*
- Policy expressed as policy axiom
  \[ r1 : \text{frank says} \]
  \[ \forall S. \text{depthead says secretary}(\text{frank}, S) \]
  \[ \supset \text{frank says open}(S, \text{weh.8117}) \]
- Policy known to Jenn, Frank, and WeH 8117
- Jenn approaches WeH 8117 with smart phone
- WeH 8117 challenges with
  \[ \text{frank says open}(\text{jenn}, \text{weh.8117}) \]
"My secretary can open my office"

- Jenn asks database (silent phone call)
  
  ? : depthead says secretary(frank, jenn)

- Database replies with signed certificate as proof
  
  x509(cdksi92899) : depthead says secretary(frank, jenn)

- Jenn assembles and sends proof
  
  r1(x509(cdksi92899))

- WeH 8117 checks
  
  r1(x509(cdksi92899)) : frank says open(jenn, weh.8117)

- Door opens
"My secretary can open my office"

- Could also relativize "my office"

\[
\forall P. \forall O. \text{depthead says office}(P, O) \supset \text{office}(P, O)
\]
\[
\forall P. \forall O. \text{office}(P, O) \supset \text{open}(P, O)
\]

- Simplified proof expression here for brevity
- Knowledge can be shared and distributed since signed
- Certificates and proofs can be cached
- Checking certificates checks expiration
Authorization Logic Implementation

- Representation in Logical Framework
  - Logic: LF signature
  - Policy: LF signature of restricted form
  - Proof: LF object
- Proof generation [Bauer, Garriss, Reiter’05]
  - Extensive caching to minimize communication
  - Distributed certifying prover
- Proof checking
  - X.509 certificate checking
  - Proof checking as LF type checking
Some Authorization Logic Issues

- Intuitionistic or classical?
- Laws for “says” modality?
- Set of logical connectives?
- Propositional or first-order or higher-order?
- Decidable?
- Monotonic?
- Temporal?
Logic Design Principles

• Proof-theoretic semantics
  [Martin-Löf’83] [Pf & Davies’01]
  • Separating judgments from propositions
  • Characterize connectives and modalities via their rules
  • Cut elimination and identity principles
  • Focusing [Andreoli’92]

• Consequences
  • Independence of logical connectives from each other
  • Intuitive interpretation
  • Amenable to meta-theoretic analysis (exploration!)
  • Open-ended design (extensibility!)
Judgments

- **Judgments** are objects of knowledge
- **Evidence** for judgments is given by deductions
- Basic judgments
  - $A \text{ true}$ — proposition $A$ is true
  - $P \text{ aff } A$ — principal $P$ affirms proposition $A$
- Logical connectives are defined by their *introduction* and *elimination* rules
- Must match in certain ways to be meaningful
- Here, truth is almost subsidiary, because affirmation expresses intent
Hypothetical Judgments

- **Hypothetical judgments** for reasoning from assumptions
  \[ J_1, \ldots, J_n \vdash J \]

- Will freely reorder assumptions
- **Hypothesis rule**
  \[ \Gamma, J \vdash J \]

- **Substitution principle**
  If \( \Gamma \vdash J \) and \( \Gamma, J \vdash J' \) then \( \Gamma \vdash J' \).

- Fixes meaning of hypothetical judgments
Implication

• Introduction rule

\[ \frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \supset B \text{ true}} \vdash I \]

• Elimination rule

\[ \frac{\Gamma \vdash A \supset B \text{ true} \quad \Gamma \vdash A \text{ true}}{\Gamma \vdash B \text{ true}} \vdash E \]
Local Soundness

- An introduction followed by any elimination of a connective can be reduced away
- Shows elimination rules are not too strong

\[
\frac{\varepsilon}{\Gamma, A \text{ true} \vdash B \text{ true}} \quad \frac{\Gamma \vdash A \supset B \text{ true}}{\Gamma \vdash A \text{ true}} \quad \frac{\Gamma \vdash A \text{ true}}{\Gamma \vdash B \text{ true}} \quad \frac{\Gamma \vdash B \text{ true}}{\Gamma \vdash B \text{ true}}
\]

\[\varepsilon' \Rightarrow_R \Gamma \vdash B \text{ true}\]

- \(\varepsilon'\) constructed by substituting \(\mathcal{D}\) in \(\varepsilon\)
- Possible by substitution principle
Local Completeness

- There is a way to apply eliminations to a compound proposition so we can reintroduce the proposition from the results.
- Shows elimination rules are not too weak.

\[
\frac{\Gamma, A \text{true} \vdash A \supset B \text{true} \quad \Gamma, A \text{true} \vdash A \text{true}}{\Gamma \vdash A \supset B \text{true} \quad \Gamma \vdash A \supset B \text{true} \quad \vdash E}
\]

- \(\mathcal{D}'\) constructed by weakening from \(\mathcal{D}\)
Truth and Affirmation

- Define *affirmation judgment* relative to truth
- If $A$ is true then any $P$ affirms $A$

$$
\Gamma \vdash A \ true \\
\Gamma \vdash P \ aff \ A
$$

- If $P$ affirms $A$, then we can assume $A$ is true, but only while establishing an affirmation by $P$

$$
\text{If } \Gamma \vdash P \ aff \ A \ \text{and } \Gamma, A \ true \vdash P \ aff \ C \\
\text{then } \Gamma \vdash P \ aff \ C
$$
Internalizing Judgments

- Implication internalizes hypothetical reasoning
- “says” modality internalizes affirmation
- Introduction rule

\[
\Gamma \vdash P \aff A \\
\Gamma \vdash (P \text{ says } A) \text{ true} \quad \text{saysI}
\]

- Elimination rule

\[
\Gamma \vdash (P \text{ says } A) \text{ true} \quad \Gamma, A \text{ true} \vdash P \aff C \\
\Gamma \vdash P \aff C \quad \text{saysE}
\]
Local Soundness

• Reduce introduction followed by elimination

\[ \Gamma \vdash P \text{aff} A \]
\[ \Gamma \vdash (P \text{ says } A) \text{ true} \]
\[ \Gamma \vdash P \text{ aff } C \]

\[ \Gamma, A \text{ true } \vdash P \text{ aff } C \]  

\[ \Gamma \vdash P \text{ aff } C \]  

\[ \Rightarrow R \quad \Gamma \vdash P \text{ aff } C \]

• \( \mathcal{E}' \) is constructed from \( \mathcal{D} \) and \( \mathcal{E} \)

• Exists by definition of affirmation
Local Completeness

- Eliminate to re-introduce

\[ D \]
\[ \Gamma \vdash (P \text{ says } A) \text{ true} \quad \Rightarrow_E \]
\[ G \]
\[ \Gamma \vdash (P \text{ says } A) \text{ true} \quad \Gamma, A \text{ true} \vdash A \text{ true} \]
\[ \Gamma, A \text{ true} \vdash P \text{ aff } A \quad \text{says } E \]
\[ \Gamma \vdash P \text{ aff } A \quad \Gamma \vdash (P \text{ says } A) \text{ true} \quad \text{says } I \]
Some Consequences

• Principals are isolated: they only share truth!
• Dependencies only from policy axioms

\[
\begin{align*}
\text{frank says} & \\
\forall S. \text{depthead says secretary}(\text{frank}, S) & \supset \text{frank says open}(S, \text{weh.8117})
\end{align*}
\]
Affirmation as Indexed Monad

- $P$-indexed family of strong monads
  - $\vdash A \supset (P \text{ says } A)$
  - $\vdash (P \text{ says } A) \supset (A \supset (P \text{ says } C)) \supset (P \text{ says } C)$
  - $\vdash (A \supset B) \supset ((P \text{ says } A) \supset (P \text{ says } B))$
  - $\vdash (P \text{ says } (P \text{ says } A)) \supset (P \text{ says } A)$

- Strong monads used in functional programming to isolate effects

- $P$ says $A$ corresponds to $\Diamond \Box A$ from *lax logic*
  [Benton, Bierman, de Paiva’98]

- Decomposes into $\Diamond \Box A$ from *modal logic* CS4
  [Pf. & Davies’01]
Other Connectives

• Judgmental foundation allows modular addition of new connectives by introductions and eliminations

• Quantifiers are also straightforward

• Some consequences:
  • $\vdash ((P \text{ says } A) \lor (P \text{ says } B)) \supset (P \text{ says } (A \lor B))$
  • $\not\vdash (P \text{ says } (A \lor B)) \supset ((P \text{ says } A) \lor (P \text{ says } B))$
  • $\vdash \bot \supset (P \text{ says } \bot)$
  • $\not\vdash (P \text{ says } \bot) \supset \bot$

• Last property is critical, since principals are not constrained in what they affirm
Cut Elimination

• How do we prove $\not\vdash (P \text{ says } \bot) \supset \bot$?
• Generalize from local soundness and local completeness to global properties
• Via cut-free atomic sequent calculus
• Show cut and identity principle are admissible
• Introduce new basic judgment
  \( A \ hyp \) — proposition \( A \) is hypothesis

• Use only on left-hand side of hypothetical
  \( A_1 \ hyp, \ldots, A_n \ hyp \vdash A \ true \) (write: \( \Delta \ \Rightarrow \ A \ true \))
  \( A_1 \ hyp, \ldots, A_n \ hyp \vdash P \ aff \ A \) (write: \( \Delta \ \Rightarrow \ P \ aff \ A \))

• Judgmental rules

\[
\begin{align*}
\Delta, a \ hyp & \Rightarrow a \ true \\
\Delta \Rightarrow A \ true \\
\Delta \Rightarrow P \ aff \ A
\end{align*}
\]
Sequent Rules

- Right rule from intro, left rule from elim
- Omit (implicit) contraction
- $J$ either $C$ true or $P$ aff $C$

\[
\begin{align*}
\Delta, A \text{ hyp } & \Rightarrow B \text{ true } & \Delta, B \text{ hyp } & \Rightarrow J \\
\Delta \Rightarrow A \supset B \text{ true } & \supset R & \Delta, A \supset B \text{ hyp } & \Rightarrow J \quad \supset L \\
\Delta \Rightarrow P \text{ aff } A & \Rightarrow R & \Delta \Rightarrow P \text{ aff } C & \Rightarrow L \\
\Delta \Rightarrow (P \text{ says } A) \text{ true } & \Rightarrow R & \Delta, (P \text{ says } A) \text{ hyp } & \Rightarrow P \text{ aff } C
\end{align*}
\]
Cut and Identity

- Cut (global soundness)
  \[ \text{If } \Delta \Rightarrow A \text{ true and } \Delta, A \text{ hyp } \Rightarrow J \text{ then } \Delta \Rightarrow J \]

- Proof by simple nested structural induction on \( A \) and the two given derivations

- Identity (global completeness)
  \[ \Delta, A \text{ hyp } \Rightarrow A \text{ true for any proposition } A \]

- Proof by simple structural induction on \( A \)

- \( \Gamma \vdash J \) iff \( \Gamma \Rightarrow J \) (from cut, with abuse of notation)
Some Easy Consequences

• Subformula property
• Immediate independence results
  • \( \not\Rightarrow \bot \text{ true} \)
  • \( \not\Rightarrow P \text{ aff} \bot \)
  • \((P \text{ says } \bot) \text{ hyp} \not\Rightarrow \bot \text{ true} \)
  • \(A \supset (P \text{ says } B) \text{ hyp} \not\Rightarrow (P \text{ says } (A \supset B)) \text{ true} \)

• Simple non-interference

\( \text{If } \Delta \text{ and } J \text{ do not mention } P, \text{ then } \) 
\( \Delta, P \text{ says } A_1 \text{ hyp}, \ldots, P \text{ says } A_n \text{ hyp } \Rightarrow J \text{ iff } \) 
\( \Delta \Rightarrow J. \)


### Reasoning About Logic and Policies

- We have formally verified cut in Twelf (proof explicitly supplied) [Pf & Schürmann’99, Pf’00, Garg’05]
- Some independence results are easily verified formally
- Conjecture: these can be proven automatically [Pf & Schürmann’98]
- Deeper reasoning about policies (= sets of axioms) is tricky
  - Requires (at least) focusing
  - Clean proof theory may enable some results
Expressive Power

- Easy
  - Groups and roles
  - Delegation of specific rights
  - Joint authorization

- Slightly more complicated (not yet verified)
  - Full delegation
  - Creating new principals
Intuitionistic vs Classical Logic

• Intuitionistic logic as logic of explicit evidence
• Sample classical, but not intuitionistic truth
  [Abadi’03]

\[(P \text{ says } A) \supset (A \lor (P \text{ says } B))\] for any \(B\)

• Classical logic is *descriptive*, arises from structure
• Intuitionistic logic is *creative*, arises from properties
• Authorization is not given explicitly by a structure, but by properties (non-interference)
Authorization Logic Issues, Revisited

- Intuitionistic or classical? (intuitionistic)
- Laws for \texttt{says} modality? (indexed family of strong monads)
- Set of logical connectives? (open-ended)
- Propositional or first-order or higher-order? (first-order)
- Decidable? (no, fragment tractable?)
- Monotonic? (yes)
- Temporal? (no)
Monotonicity

- Nonmononicity dubious in distributed setting
- Instead, for access revocation:
  - Short-lived certificates
  - `notRevoked` predicate
  - External reasoning about time
- Ephemeral capabilities (future work)
  - Digital rights management
  - Electronic payment
  - Bounded delegation
  - Via *linear connectives in authorization logic?*
Most Closely Related Work

• [Abadi, Burrows, Lampson, Plotkin’93] propositional, axiomatic, rich calculus of principals

• [Appel & Felten’99] [Bauer’03] (PCA) classical, higher-order, no analysis of modalities

• [De Treville’02] (Binder) datalog, decidable, modality not classified

• [Rueß & Shankar’03] (Cyberlogic) intuitionistic, unjustified modal laws, semi-axiomatic style, more ambitious scope (protocols), proof-carrying

• [Abadi, LICS 2003] structured overview, further references
Desiderata Revisited

- Expression, Enforcement, Exploration (EEE)
  - Expressive policy language
  - Simple enforcement of policies
  - Feasible reasoning about policies

- Extensibility

- Small trusted computing base

- Smooth integration of authentication

- Work with distributed information
Conclusion

- Design of authorization logic as modal logic
  - Judgmental, constructive, open-ended, modular
  - Affirmation as indexed strong monad
  - Basic cut elimination formally verified
- Next
  - Extend verification to more connectives
  - Stronger non-interference properties
  - Cell-phone implementation (currently higher-order logic)
- Eventually:
  - Linear authorization logic for ephemeral capabilities (digital rights, electronic payments, bounded delegation)