Constructive Access Control: Revisited?

Valeria de Paiva
Intelligent Systems Lab
PARC

(Joint work with Jessica Staddon, CSL)

parc
Palo Alto Research Center

Outline

- Motivation: access control must be logic…
- Background
- Basic framework
- A new system?
- Discussion & applications

Caveat: no expert, a talk to logicians interested in the problem…

Why the buzz about access control?

- Ubiquity of computing and growth of the Internet turned Information Security into a central area of research in computer science.
- Many areas within Information Security. For logicians there's considerable work on logical methods for access control.
- For example:
  - Abadi et al, 1993, **Abadi, 2003**, Abadi 2006
  - Garg et al, 2006
  - Garg, Pfenning 2006
  - Garg, Abadi, 2008
    - » Thanks Martin and Deepak!

Access control in current practice (according to Abadi)

- Access control is pervasive
  - applications
  - virtual machines
  - operating systems
  - firewalls
  - doors
  - …
- Access control seems difficult to get right.
- Distributed systems make it harder.

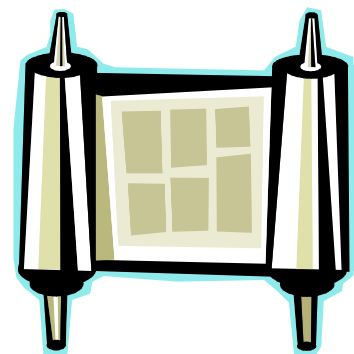
What is Access Control?

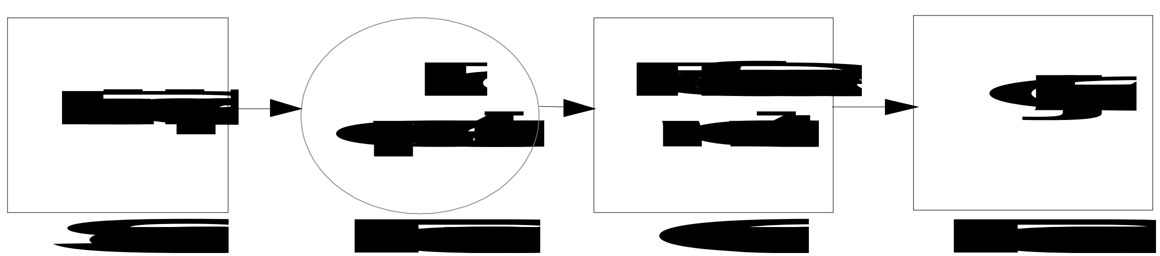
- In computer security, access control consists in deciding whether an agent that requests some action should have his request granted or not.
- Decisions are based on access control policies, the combination of several policies at different layers and from different entities.
- A single policy may be easy to understand e.g. user **Valeria** may want to delete **file1** and if she owns the file the **admin** should allow it.
- But the consequences of even a single policy can get complicated, when there are many principals, many roles, many resources, delegation, revocation, etc.

The access control model

Elements:

- Resources
- Requests
- Sources for requests, called principals
- A reference monitor to decide on requests
- Control policies





General theories and systems

- Over the years, there have been many theories and systems for access control.
  - Logics
  - Languages
  - Infrastructures (e.g., PKIs)
  - Architectures
- They aim to explain, organize, and unify access control.
- We’re interested in logics and languages…

Access Control needs logic?

"*Although access control may sometimes seem conceptually straightforward, it is both complex and error-prone in practice*. […] *One may hope that logic would provide a simple, solid, and general foundation for access control, as well as methods for designing, implementing, and validating particular access control mechanisms*. *In fact, although logic is not a panacea, its applications in access control have been substantial and beneficial.*" M. Abadi, Invited Address, LICS 2003

parc Palo Alto Research Center

Access control needs logic

- We need to combine access control policies, have groups of principals, revocation, delegation, roles, etc.
- Things can get very complicated. There can be gaps, inconsistencies, ambiguity, loopholes, obscurity.
- Systems can be easy to break and security is endangered.

On the other hand…

(Constructive) Logic can:

- Express policies
  - Admin says

    owns (Valeria, file) -> may_delete(Valeria, file)
- Express authorization questions
  - Does Valeria have a proof of the proposition

    Admin says may_delete(Valeria, file)?
- Logical proofs allow us:
  - Construct evidence (assemble proof)
  - Verify evidence (verify proof)
  - Reason from assumptions (given credentials)

Logics for Access Control

- Encode and reason within policies
- Analyze policies (reason about them)
  - Express (and reason about) private knowledge?
- Prove properties of policies, check for unintended consequences. Enforce policies?
- Proofs hard to construct, easy to verify
  - Lead to Proof Carrying Authorization Appel&Felten, Bauer
- PCA insight :

the user/ principal wanting access must construct a proof, the server will simply check the proof to grant access

uses higher-order logic, can we make it simpler?

Logics of Access Control

- Several systems proposed and studied.
- Traditionally classical **modal** logics with extra constructs (Abadi et al 1993)
- Garg&Pfenning(2006) have proposed a **constructive** lax logic of access control, non-interference
- Abadi (2006) has proposed a lax logic based system DCC, non-interference
- Garg et al(2006) have proposed a “linear” logic for access control, credentials are resources
- Garg&Abadi(2008 to appear) have four systems based on lax logic

Background1: Principals

- A principal is any user, machine, program, organization that
  - Either makes requests, or
  - Makes statements (policies)
- Examples:
  - Humans: Alice, Bob, Charlie, ...
  - Users: 500, 501, admin, ...
  - Programs: MSWord, Acrobat Reader, ...
  - Organizations: CMU, SRI, ACM, Wells-Fargo...
  - Public keys: 0xaf5436, 0x123458

Background2: “A says s”

Taking Garg&Abadi (GA08) as basic reference

- Basic construct operator “says”: applied to principal A and formula s, “A says s”.
  - Abstracts away from implementation concerns
- “A says s” means intuitively that A asserts or supports s, e.g. “A says delete-file1”.
- Different access control logics have subtly different meanings for “says”.
- Note similarity to “K attests A” in cyberlogic, where K is (has to be?) a public key, A is a formula

Background3: "speaks for"

- Operator "speaks for", applied to principals **A** and **B,**
  A => B
- This is read "**A** speaks for **B**" and intuitively means that if **A** says s then **B** says s, for all s.
- In particular if $\mathbf{K_{Alice}}$ is the public key for Alice we have $\mathbf{K_{Alice}}$ **=> Alice.**

  also if S a server then S => Alice, if S is acting for Alice
- Different access control logics have subtly different meanings for "speaks for"
- Not fine-grained enough?
- (Similar to cyberlogic's delegation?)

Which logic of access control?

- Intuitionistic basis, as we want
  – a Curry-Howard isomorphism,
  – evidence instead of truth
  – use proofs as witnesses for PCA
- Have a collection of principals **A, B,..**
- How do we represent logically the constructs for access control?
- All recent work mentioned uses an indexed collection of **lax modalities**

What's a lax modality?

- A modality is an unary operator acting on propositions
- Curry(1952) a possibility modality that half-behaves like a necessity one.
- Like possibility, twice the modality implies it once. But like necessity as it satisfies distribution over implication.
- Also known as computational logic, CL, (Benton, Bierman, de Paiva, JFP 1998)
- Properties:

  s → A says s

  A says A says s → A says s

  A says (s→t) → (A says s) → (A says t)

Why lax modalities?

- Need to model “A says s”
- “says” has some characteristics of possibility:

  if “A says (A says s)” then “A says s”,

  if “A says (s->t)” then “A says s-> A says t”
- Lax modalities buy you non-interference (Abadi06, GargPfenning06)
- Lax modalities buy you “hand-off axiom”: if A says that B speaks for A then B does speak for A (Abadi06)
- Lax modality well-understood logic type theory

parc Palo Alto Research Center

How to do lax modalities?

- Different proof systems: Moggi89,de Paiva et al 98, Mendler&Fairtlough97
- Garg&Pfenning: ‘judgemental’ logic (2001)
- Based on Martin-Loeuf’s ideas: intro and elim rules plus cut elim are the meaning of connectives
- Works for S4-style connectives, dual-sized sequents (e.g. linear logic exponentials)
- Can we do less powerful/less symmetric modalities?

Why not lax modalities?

- Axiom (s -> A says s) means every principal says s, if s is true
  – Difficult to believe that principals are that ideal
- Similarly, “speaks for” too strong
- Alice would like to make sure that Bob speaks for her in certain circumstances, not for all s.
- Maybe can use a simple K constructive modality for “says”…

A new system?

- Caveat: work not really done…
- But Curry-Howard Iso for Basic Modal Logic, (Bellin, de Paiva, Ritter, 2001)
- Bug in published version, being corrected and extended now
  - Thanks to Kakutani (2006) for correcting it!
- Type theory, semantics in place:
  - Normalization, subject reduction, soundness&completeness, internal language too
- Non-interference works too, “hand off”?

Extensions

- Garg: linear logic to deal with credentials that are consumable resources
  - Apparently proof-theory done, implementation is the problem
  - Garg et al 06, Bauer et al 06
- Garg et al: temporal aspects of security in the works
  - I also want my versions with and without linear basis
  - Constructive temporal logics in the market not good

Applications?

- A bit of unifying glee: 1995 proposal on logics of authentication
- PCA for less expressive logics
  - Grey project at CMU interesting, but it would be nice if it could be simpler, Manifest Security?
- Access control for multiple enterprise repositories:
  - What if our principals were the parties that need to cooperate when someone is buying a house?
  - Can our access control theories help out?
  - Some Stanford/PORTIA work on this direction

Conclusion

- Logic clearly useful for access control
- Multiple applications and opportunities
- More work required on trade-offs between logical systems, automation, etc
- Innovative applications may send the formalism into totally different directions

Thank you

Questions?

References

- **Manifest Security for Distributed Information** Karl Crary, Robert Harper, Frank Pfenning 2006
- Garg&Abadi08, Garg&Pfenning06, Garg et al 06
- **PCA** Appel&Felten 99, Bauer’s thesis 03

A calculus for access control

[Abadi, Burrows, Lampson, and Plotkin, 1993]

- A simple notation for assertions
  - A says s
  - A speaks for B (sometimes written A ⇒ B)

- With logical rules
  - ⊢ A says (s → t) → (A says s) → (A says t)
  - If ⊢ s then ⊢ A says s.
  - ⊢ A speaks for B → (A says s) → (B says s)
  - ⊢ A speaks for A
  - ⊢ A speaks for B ∧ B speaks for C → A speaks for C

Enforcing policies?

- An access control policy can be presented as a logical theory in an access control logic
- A principal is granted access to a resource if there is a formal proof that the principal is authorized the use of the resource according to the accepted policy
- Constructivity buys you PCA?