Logical Foundations for Access Control

프로그래밍언어연구회 여름학교 2004 부산대

신승철@동양대

2004년 8월 12일

Logic of Talk

[**Definitions**] Security and Access Control

[**Lemmas**] Short review of Modal Logic

[**Theorem**] Modal Logic for Access Control

[**Corollary**] References for going further

What is Security?

Security is the protection of a person, property or organization from an attack. There are people who have distorted motivations to perform such attacks. The types of protection include prevention, response and preemptive attacks.

In this talk
security is the protection of a system, information or services from an attack. There are systems that have vulnerability to attacks. The types of protection include authentication, authorization and auditing.

Software security is about how we can devise security mechanisms for software systems and how we can make sure that software systems guarantee security policies.

What is Security?

Security: prevent bad things from happening

- Confidential information leaked
- Important information damaged
- Critical services unavailable
- Clients not paying for services
- Money stolen
- Improper access to physical resources
- System used to violate law
- Loss of value

or at least make them less likely

Views of Security problems

- What is being protected (and from what)
- How it is being protected (access control, cryptography, ...)

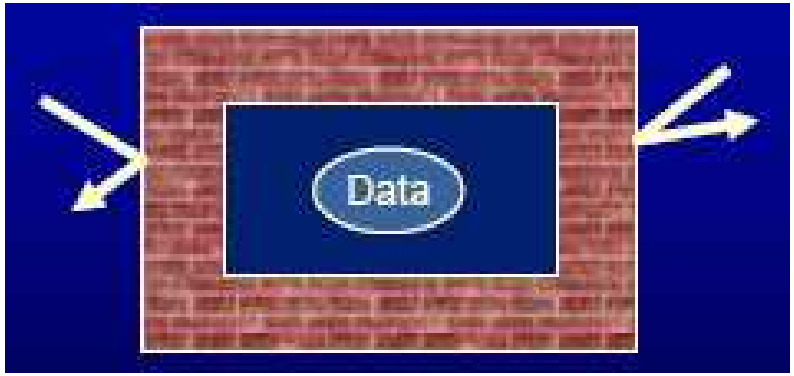
Policy vs. Mechanism

A Policy is an instance of Security properties

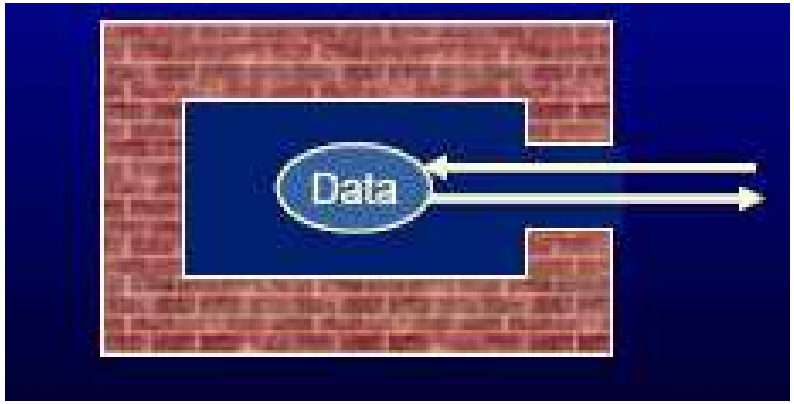
- Confidentiality
- Integrity
- Availability
- Privacy and anonymity
- No repudiation

Confidentiality

Integrity



Availability



Security Mechanism: Authentication

Is this an authentic request from principal p?

What principal p is system acting on behalf of?

Using Passwords, biometrics, certificates ...

Security Mechanism: Authorization

Is principal p authorized to perform action A?

. Access control mediates actions by principals

. Example: file permissions (ACLs)

Security Mechanism: Auditing

For after- the- fact enforcement, need to know what happened: auditing . Audit log records security- relevant actions (who, what, when)

In classic systems security

Authorization + Authentication + Audit = "The gold (Au) standard"

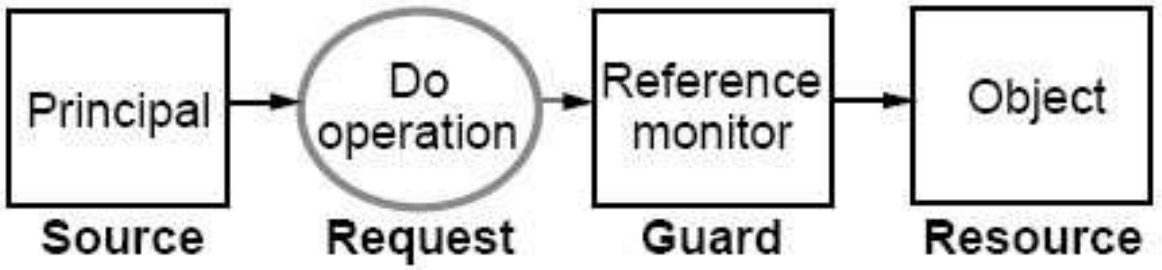
A fourth kind of mechanism: program analysis and verification

Hence software security is about how we can make security mechanisms for software systems and how we can make sure that a given software guarantees security policies.

"Access Control Model"

Elements:

- Objects or resources
- Requests
- Sources for requests, called principals
- A reference monitor to decide on requests



"Authentication vs. Access Control"

Access Control (Authorization):

- Is principal A trusted on statement S?
- If A requests S, is S granted?

Authentication:

- Who says S?
- Who is principal A on behalf of?

Access Control Matrix

objects / principals	file1	file2	file3	file4
user1	rwx	rw	r	x
user2	r	r		x
user3	r	r		x

Access Control Lists and Capabilities

Two strategies (often combined) to implement Access Control

- ACL: a column of an access control matrix, attached to an object
- Capability: a row of an access control matrix, given to a principal

objects / principals	file1	file2	file3	file4
user1	rwx	rw	r	x
user2	r	r		x
user3	r	r		x

The principle of complete mediation

Every access to every object is checked.

This principle can be enforced in several ways:

- The OS intercepts some of the subject's requests. The hardware catches others. (E.g., as in Unix.)
- A software wrapper/interpreter intercepts some of the subject's requests. (E.g., as in the JVM.)

Complicated Authorization

- Conjunctions
- Groups
- Roles
- Delegations
- .....

Conjunctions

- Sometimes a request should be granted only if it is made jointly by several principals.
- A conjunction may or may not be made explicit in the access policy.

Groups and roles

- Principals can be organized into groups.
- Principals can play roles.
- These groups and roles may be used as a level of indirection in access control.
  E.g., any member of a group G may access a file f.

More on objects and operations

- Objects and operations may also be put in groups, e.g.,
  - all company files,
  - all read operations on an object.
- Sometimes operations should be bundled, e.g.,
  - read a patient's record,
  - write a log record.

Programs

- Programs may be principals too. But then:
- we need to deal with call chains,.e.g.,
  - applet on browser on OS,
- we still need to connect programs to other principals.
  - who write them or edit them,
  - who provide them,
  - who install them,
  - who call them.

Issues

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

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 often aim to explain, organize, and unify access control.
- They may be intellectually pleasing.
- They may actually help.

Algorithmic analysis [starting with Harrison, Ruzzo, and Ullman, 1976]

- A system has finite sets of rights and commands.
- A configuration is an access control matrix.
- A command is of the form "if conditions hold, perform operations" (with some parameters).
  - The conditions are predicates on the matrix.
  - The operations add or delete rights, principals, and objects.

```
command CONFER_r (owner, friend, file)
  if own in (owner, file)
  then enter r into (friend, file)
end
```

- Safety means that untrusted subjects cannot access a resource in any reachable state: it is undecidable (in general).

Algorithmic analysis (cont.)

[in particular, Li, Winsborough, and Mitchell, 2003]

- Not all interesting problems are undecidable!
- Containment problem:
  In every reachable state, does every principal that has one property (e.g., has access to a resource) also have another property (e.g., being an employee)?
- For different classes of systems, this problem is decidable.

Formal verification

A formally verified security kernel is widely considered to offer the most promising basis for the construction of truly secure computer systems at least in the short term. A number of kernelized systems have been constructed and various models of security have been formulated to serve as the basis for their verification.Despite the enthusiasm for this approach there remain certain difficulties and problems in its application ..... (Rushby, 1981)

A logic from matrices

- An access control matrix may be represented with a ternary predicate symbol may-access.
- The setting may be a fairly standard, classical predicate calculus.
- We may then write formulas such as:
  may-access(Alice, Foo.txt, Rd)
  and rules such as:
  may-access(p, o, Wr) $\Rightarrow$ may-access(p, o, Rd)

A logic from matrices: questions

- Does this really help?
  - In describing policies?
  - In analyzing policies?
- We may need many more constructs and axioms for representing security policies.
  - may-jointly-access(p,q,o,r)
  - owns(p,o)
  - ..... (When are we done?)

Access control in distributed systems

Many characteristics of distributed systems make access control harder:

- size
- faultiness
  e.g., revocation messages may get lost
- heterogeneity
  e.g., of communication channels of protection mechanisms
- autonomy and lack of central administration and therefore of central trust
- .....

An approach

- Calculus of Principals: A notation for representing principals and their statements, and perhaps more:
  - objects and operations,
  - trust,
  - channels,
  - .....
- Doxastic Modal Logic: a logic for reasoning authorization

Basic modal logic

Alphabet :

- A set of propositional letters p, q, ...
- Propositional connectives $\neg$ and $\wedge$ and constant true $\top$
- modality $\Box$

Well-formed formula ϕ, ψ :

$$p \mid \top \mid \neg\phi \mid \phi \wedge \psi \mid \Box\phi$$

"Relational structures"

A *frame* for the basic modal language $\mathfrak{F} = (W, R)$
where W is a non-empty set of possible worlds
and R is a binary relation on W.

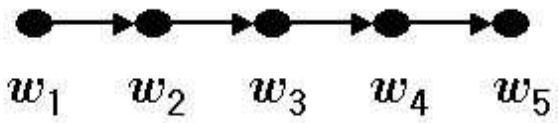
A *model* for the basic modal language $\mathfrak{M} = (\mathfrak{F}, V)$
where $\mathfrak{F}$ is a frame
and V is a valuation function $\Phi \longrightarrow \mathcal{P}(W)$.

"Satisfaction"

$$
\begin{array}{rll}
\mathfrak{M}, w \vDash p & \text{if} & w \in V(p), \text{ where } p \in \Phi \\
\mathfrak{M}, w \vDash \top & \text{if} & \text{always} \\
\mathfrak{M}, w \vDash \neg\phi & \text{if} & \mathfrak{M}, w \nvDash \phi \\
\mathfrak{M}, w \vDash \phi \wedge \psi & \text{if} & \mathfrak{M}, w \vDash \phi \quad \text{and} \quad \mathfrak{M}, w \vDash \psi \\
\mathfrak{M}, w \vDash \Box\phi & \text{if} & \text{for every } v \in W \text{ such that } Rwv, \text{ we have } \mathfrak{M}, v \vDash \phi
\end{array}
$$

"예제 1"

$\mathfrak{F} = (\{w_1, w_2, w_3, w_4, w_5\}, R)$ where Rw_iw_j iff $j = i + 1$:

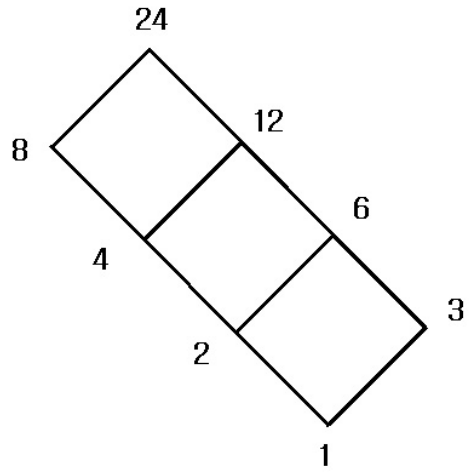


$V(p) = \{w_2, w_3\}$
$V(q) = \{w_1, w_2, w_3, w_4, w_5\}$
$V(r) = \varnothing$

$\mathfrak{M}, w_1 \vDash \Box\Box p$
$\mathfrak{M}, w_1 \nvDash \Box\Box p \rightarrow p$
$\mathfrak{M}, w_2 \vDash \Box(p \wedge \neg r)$
$\mathfrak{M}, w_1 \vDash q \wedge \Box(q \wedge \Box(q \wedge \Box(q \wedge \Box q)))$

"예제 2"

$\mathfrak{F} = (\{1, 2, 3, 4, 6, 8, 12, 24\}, R)$ where Rxy iff $x \neq y$ and x divides y:



$V(p) = \{4, 8, 12, 24\}$, $V(q) = \{6\}$

$\mathfrak{M}, 6 \models \Box p$
$\mathfrak{M}, 2 \not\models \Box p$
$\mathfrak{M}, 2 \models \Box(q \vee \Box p)$

"R-accessible Possible worlds"

$$\mathfrak{F} = (W, R)$$

$$\mathfrak{M} = (\mathfrak{F}, V)$$

When $R = W \times W$, $\quad (W, R, V), w \vDash \Box\phi$ if $\forall v \in W.(W, R, V), v \vDash \phi$

$\vdots$

$\vdots$

When $R = \varnothing$, $\quad (W, R, V), w \vDash \Box\phi$ if always

"System **K**"

The axioms of **K**:

- propositional tautologies
- $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$

The rules of proof of **K**:

- Modus ponens: given ϕ and $\phi \rightarrow \psi$, prove ψ
- Uniform substitution: given ϕ, prove θ, where θ is obtained from ϕ by uniformly replacing proposition letters in ϕ by arbitrary formulas.
- Necessitation: given ϕ, prove $\Box\phi$

Doxastic modal logic

Alphabet :

- A set of propositional letters p, q, ...
- Propositional connectives $\neg$ and $\wedge$ and constant true $\top$
- multi-modality $\Box_A$

Well-formed formula ϕ, ψ :

$$p \mid \top \mid \neg\phi \mid \phi \wedge \psi \mid \Box_A \phi$$

For multi-modality

$\Box_A \phi$ or $[A]\phi$: A believes ϕ or A says ϕ

where A is taken from some index set.

"Multi-Relational Structures"

A *frame* for the doxastic modal language $\mathfrak{F} = (W, \{R_A\})$
where W is a non-empty set of possible worlds
and R_A is a binary relation, associated with A from an index set P, on W.

A *model* for the doxastic modal language $\mathfrak{M} = (\mathfrak{F}, V)$
where $\mathfrak{F}$ is a frame
and V is a valuation function $\Phi \longrightarrow \mathcal{P}(W)$.

"Satisfaction again"

$\mathfrak{M}, w \vDash p$ if $w \in V(p)$, where $p \in \Phi$

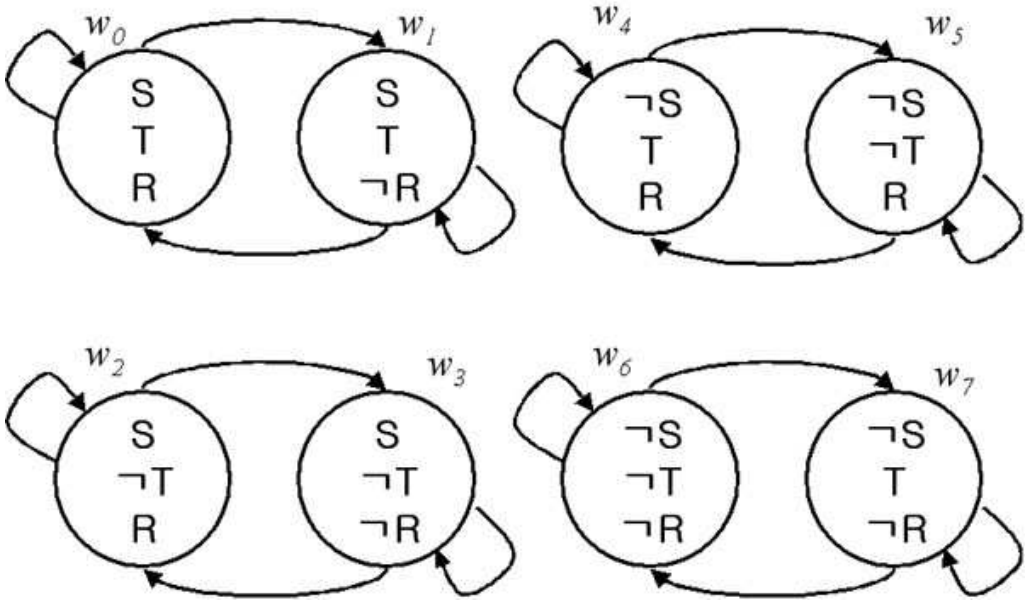
$\mathfrak{M}, w \vDash \top$ if always

$\mathfrak{M}, w \vDash \neg\phi$ if $\mathfrak{M}, w \nvDash \phi$

$\mathfrak{M}, w \vDash \phi \wedge \psi$ if $\mathfrak{M}, w \vDash \phi$ and $\mathfrak{M}, w \vDash \psi$

$\mathfrak{M}, w \vDash \Box_A \phi$ if for every $v \in W$ such that $R_A w v$, we have $\mathfrak{M}, v \vDash \phi$

A model of eight worlds



S: Alice is in summer school

¬S: Alice is in Hae-un-dae

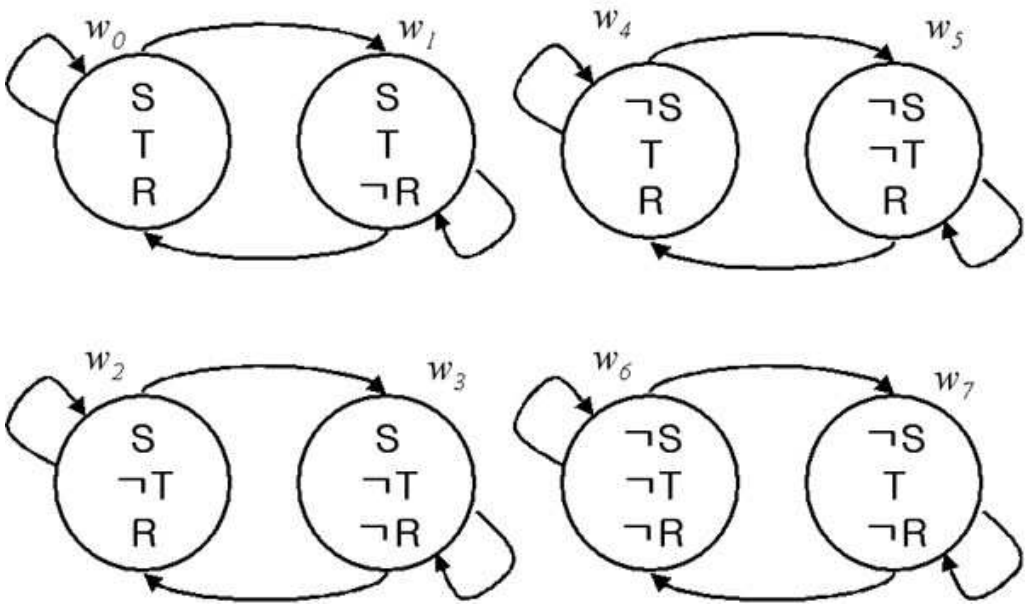
T: Talk is funny

¬T: Talk is boring

R: It is raining in Hae-un-dae

¬R: It is sunny in Hae-un-dae

A model of eight worlds



$\mathfrak{M}, w_0 \models A$ believes T

$\mathfrak{M}, w_0 \not\models A$ believes R

$\mathfrak{M}, w_0 \not\models A$ believes $\neg R$

Compound Principals

Principal expressions $\mathcal{A}, \mathcal{B}$:

- primitive principals $A \in P$
- $\mathcal{A} + \mathcal{B}$
- $\mathcal{A}|\mathcal{B}$

$\mathcal{R} : P \to \mathcal{P}(W \times W)$

- $\mathcal{R}(A) = R_A$
- $\mathcal{R}(\mathcal{A} + \mathcal{B}) = \mathcal{R}(\mathcal{A}) \cup \mathcal{R}(\mathcal{B})$
- $\mathcal{R}(\mathcal{A}|\mathcal{B}) = \mathcal{R}(\mathcal{B}) \circ \mathcal{R}(\mathcal{A})$

$$\mathfrak{M}, w_0 \vDash \mathcal{B}|\mathcal{A} \text{ says } \phi$$

$$\mathfrak{M}, w \vDash \phi \text{ for all } w \text{ such that } w \in \bigcup_{w' \in \mathcal{R}(B)(w_0)} \mathcal{R}(A)(w')$$

Calculus of Principals

- multiplicative semilattice semigroups (with $+$ and $|$)
- Algebra of binary relations over a set with union and composition
- not finitely axiomatizable - the set of valid formulas is not recursive
- decidability and tractability from restricted forms of formulas

"System $\mathbf{K}_n$"

The axioms of $\mathbf{K}_n$:

- propositional tautologies
- (K) A says $(\phi \rightarrow \psi) \rightarrow (A \text{ says } \phi \rightarrow A \text{ says } \psi)$

The rules of proof of $\mathbf{K}_n$:

- Modus ponens: given ϕ and $\phi \rightarrow \psi$, prove ψ
- Uniform substitution: given ϕ, prove θ, where θ is obtained from ϕ by uniformly replacing proposition letters in ϕ by arbitrary formulas.
- Necessitation: given ϕ, prove A says ϕ for every A

For the calculus of principals

Definitions:

- $(A + B)$ says $\phi \equiv (A \text{ says } \phi) \wedge (B \text{ says } \phi)$
- $(B|A)$ says $\phi \equiv B$ says $(A$ says $\phi)$

Axioms:

- $(A + B) + C = A + (B + C)$
- $A + B = B + A$
- $A + A = A$
- $(A|B)|C = A|(B|C)$
- $A|(B + C) = (A|B) + (A|C)$
- $(A + B)|C = (A|C) + (B|C)$

"speaks for" relation

- $B \Rightarrow A$ is read "B speaks for A"
- Definition: $(B \Rightarrow A) \equiv (B = B + A)$

- Theorem: $(B \Rightarrow A) \rightarrow ((B \text{ says } \phi) \rightarrow (A \text{ says } \phi))$
- Theorem: $(B \Rightarrow A) \wedge (C \Rightarrow B) \rightarrow C \Rightarrow A$

- Axiom: $(A \Rightarrow B) \rightarrow ((A + C) \Rightarrow (B + C))$
- Axiom: $(A \Rightarrow B) \rightarrow ((A|C) \Rightarrow (B|C))$
- Axiom: $(A \Rightarrow B) \rightarrow ((C|A) \Rightarrow (C|B))$

$$C|B \Rightarrow A \quad \text{vs.} \quad \forall \phi.\, C \text{ says } B \text{ says } \phi \rightarrow A \text{ says } \phi$$

"speaks for" semantics

$$\begin{aligned} & \mathfrak{M}, w \vDash B \Rightarrow A \\ \text{iff} \quad & \mathcal{R}(B) = \mathcal{R}(B + A) = \mathcal{R}(B) \cup \mathcal{R}(A) \\ \text{iff} \quad & \mathcal{R}(A) \subseteq \mathcal{R}(B) \end{aligned}$$

So "speaks for" is independent of the world w

$$\mathfrak{M} \vDash B \Rightarrow A \quad \text{iff} \quad \mathcal{R}(A) \subseteq \mathcal{R}(B)$$

Access Control Lists

Definition: A controls $s \equiv (A \text{ says } s) \rightarrow s$

From assumptions:

- a primitive proposition s_{write}: it is okay to write to the file O_1
- $B \Rightarrow A$
- A controls s_{write}

if given B says s_{write}, we have "it is okay to write to the file O_1".

ACL$(O_1) = \{A$ controls s_{read}, A controls s_{write}, B controls $s_{read}\}$

Tractable decision making

Building proofs for an agent's access takes exponential time when assumptions include speaks-for formulas with arbitrary combinations of $\wedge$ and $|$ operators.

We define two high-level operators, "as" and "for", in terms of the lower-lever operators. Each operator is designed to reflect an idiomatic usage pattern of the calculus. These operators can restrict how the lower-level operators combine, and exploit characteristics such as associativity and idempotence.

References for Applications and Extensions

- Security in an operating system [Wobberet al.]
- An account of security in JVMs[Wallachand Felten]
- Web access control system [Bauer et al.]
- PolicyMaker and KeyNote[Blaze et al.]
- SDSI [Lampson and Rivest]
- SPKI [Ellison et al.]
- D1LP and RT [Li et al.]
- SD3 [Jim]
- Binder [DeTreville]
- XrML2.0

Conclusion

- Security has policies and mechanisms.
- Access control problem can be more complicated.
- Logic is not a panacea but so useful.
- Calculus of principals + modal logic
- Tractable decision making from restricted forms of formulas