A Theory for Comparing the Expressive Power of Access Control Models

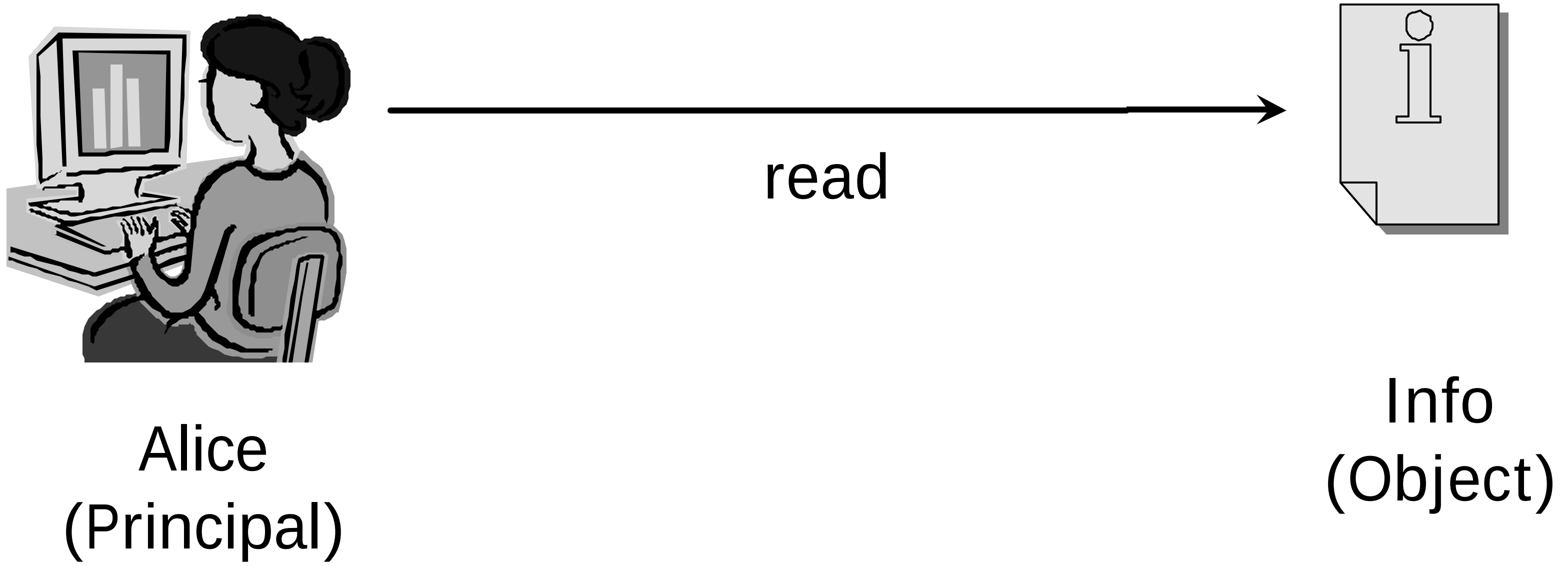
Mahesh Tripunitara Ninghui Li

Computer Science and CERIAS
Purdue University

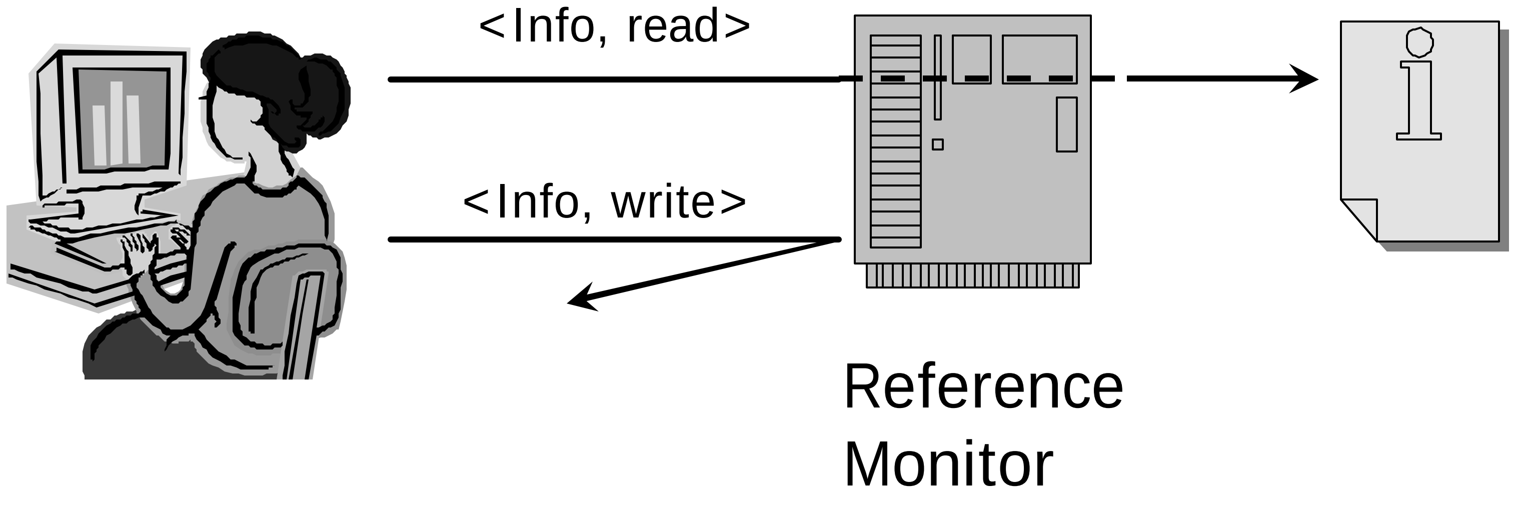
Contents

Contents

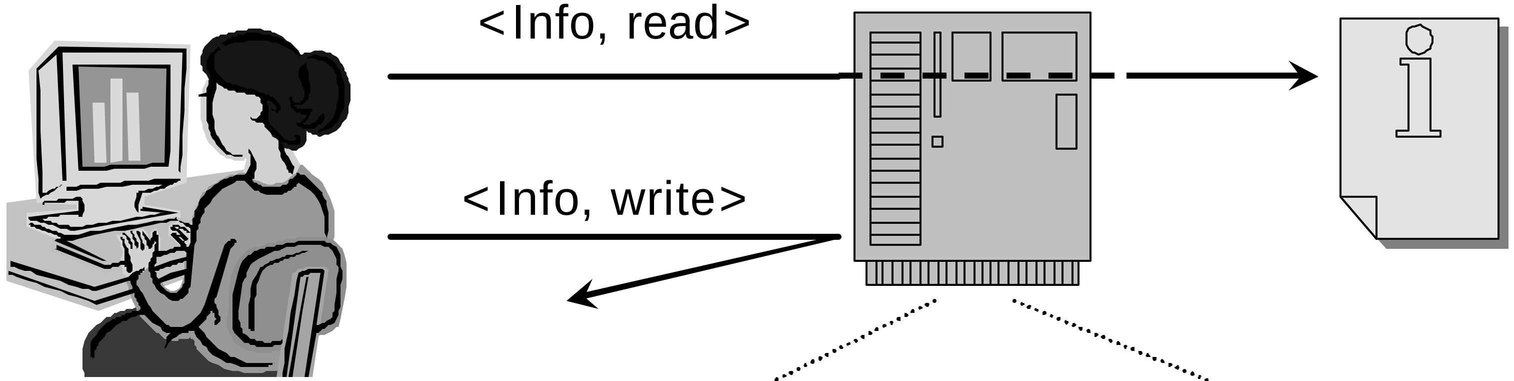
Access Control



Access Control (contd.)



Access Control (contd.)



	Alice	Info	More Info	
Alice		read	own	
Bob	control	own, write		

Protection State

- In our example, s is characterized by: $<P_s, O_s, R_s, M_s[\ ]>$
  - Specifies *access control model*
- Can query the state:
  - q_1 = “$\sigma \in P$”
  - q_2 = “$\omega \in O$”
  - q_3 = “$r \in M[\sigma,\omega]$”
- Entailment – whether query is true:
  - s ? q_3 iff $\sigma \in P_s$? $\omega \in O_s$? $r \in M_s[\sigma,\omega]$

State can change

	Alice	Info
Alice		read
Bob	control	own, write

	Alice	Info
Alice		own, read
Bob	control	write

State Change Rules

```
createObject(i,o)
    create object o
    enter own into M[i,o]
```

```
destroyObject(i,o)
    if own $\in$ M[i,o]
        destroy object o
```

```
transferOwn(i,p,o)
    if own $\in$ M[i,o]
        enter own into M[p,o]
        remove own from M[i,o]
```

```
grant_r(i,p,o)
    if own $\in$ M[i,o]
        enter r into M[p,o]
```

Systems and Schemes

- Access control system: <s, c, Q, ?>
- Access control scheme: <S, C, Q, ?>
  - $s \in S$
  - $c \in C$

- The above scheme is Strict DAC with Change of Ownership (SDCO)
  - sub-scheme of the Graham-Denning scheme

Another Scheme – ARBAC97

- s = <UA, PA, RH, AR>
- c :

assignUser	revokeUser
assignPermission	revokePermission
addToRoleRange	removeFromRoleRange
assignAsSenior	removeAsSenior

- Q : (1) $<u,r> \in$ UA;

  (2) $\exists$ u s.t. $<u,r> \in$ UA;

  (3) $\exists$ r s.t. $<u,r> \in$ UA;

  (4-6) for permissions;

  (7) $<r_1, r_2> \in$ RH;

  (8) $\exists$ r_1,r_2 s.t. $<r_1,r_2> \in$ RH ? $<u,r_1> \in$ UA ? $<p,r_2> \in$ PA

Other Examples of Schemes

- The HRU scheme (based on the access matrix model).
- Various DAC schemes (based on the access matrix model).
- MAC schemes.
- Other RBAC schemes.
- The RT[?, n] trust management scheme.

Contents

Comparison

- How does SDCO compare to ARBAC97?
- Why is this an important question?
  - can scheme B “represent” every security policy that scheme A can?
- On what basis do we compare?
  - Or, how do we formalize “represent policies”?
- Note: straightforward extension from schemes to models

Examples of Policy Questions

- Can (presumably untrusted) Alice get read access to file, f ?
- Does (administrator) Bob always have access to a configuration file?
- Does someone always have access to the building ?
- Is every object owned by exactly one principal?
- Can anyone other than Dorothy get access to the resource r ?

Our Theory: Introduction

- Does there exist a mapping from scheme A to B with relevant properties?
  - Or, can B “simulate” A?
  - Mapping should be security preserving.
  - Efficiency is not necessarily relevant.
    - But if the mapping is efficient, there is a useful implication.

Security-Preserving Mapping

- For B to be at least as expressive as A:
  - Identify security properties in A and B (e.g., safety, availability, mutual exclusion, liveness).

  - Does there exist a mapping, m from A to B, and p_A to p_B such that: $a \in A$ has p_A iff $m(a) = b \in B$ has p_B.

Questions...

- How do we represent properties of interest?
    - Answer: queries

- How do we determine whether a system satisfies a property?
    - Answer: security analysis

Security Analysis

- Access Control Scheme: <S, C, Q, ?>
- Given a system a = <s_0, c, Q, ?>, we ask:
  - ∃ reachable s_1, such that s_1 ? q?
  - ∀ reachable s_1, does s_1 ? q?
- Can check several interesting properties.

- Other kinds of questions are possible and meaningful for security – future work.
  - Example: Chinese-Wall policies

Back to Security-Preserving Mapping

- m: $(S_A \times C_A)$? Q_A ? $(S_B \times C_B)$? Q_B
- m is security preserving, if it maintains results of security analyses.

- If m is efficient, we can use analysis in B for analysis in A.

- Comparison to NP-hardness reductions.

Strongly Security Preserving Mapping

- m is strongly-security preserving, if it maintains results of compositional security analyses.
  - Compositional security analysis: allows a propositional logic formula of queries.
  - Strongly security preserving implies security preserving.

Return to our Example: SDCO

- Suppose s satisfies: $\forall \omega \in O_s$, $\exists$ exactly one $\sigma \in P_s$ such that own $\in M_s[\sigma, \omega]$

```
createObject(i,o)
   create object o
   enter own into M[i,o]

destroyObject(i,o)
   if own ∈ M[i,o]
      destroy object o

transferOwn(i,p,o)
   if own ∈ M[i,o]
      enter own into M[p,o]
      remove own from M[i,o]

grant_r(i,p,o)
   if own ∈ M[i,o]
      enter r into M[p,o]
```

SDCO (contd.)

- c maintains invariant.
- Let $\omega \in O_s$ with owner σ_1.
  - Can reach a state in which σ_2 is the owner.
  - Cannot reach state, s', in which more than one owner, or no owner (when $\omega \in O_{s'}$)
- Can represent each of the above as formula of queries from Q.

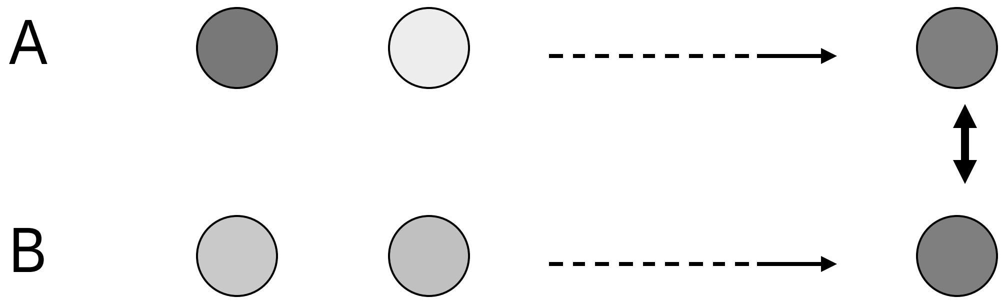
Results for SDCO and ARBAC97

- There exists a security preserving mapping from SDCO to ARBAC97.

- There exists no strongly-security preserving mapping from SDCO to ARBAC97.
    - Any ARBAC97 system must enter “extra” or “bad” state that violates invariant in trying to maintain it.

Contents

More Usable Definitions



- Are there corresponding reachable states under m?
  - Reduction: for each query.
  - State-matching reduction: for all queries.

More Usable Definitions (contd.)

- Necessary and sufficient conditions for
  - security-preserving mapping: reduction
  - strongly security-preserving mapping: state-matching reduction

- Reduction: $A =_R B$
- State-Matching Reduction: $A =_S B$

Proof Strategy

- If there exists (state-matching) reduction:
  - By construction of m
  - Show properties are satisfied
- If there exists no (state-matching) reduction:
  - By contradiction
  - Find system in A and reachable state, s_a such that for any corresponding system in B, in reaching $m(s_a)$, we have to traverse a “bad” state.

Results

- SDCO $=_R$ ARBAC97 scheme.
- SDCO $?_S$ ARBAC97 scheme.
- URA97 scheme $=_S$ RT[? , n] scheme.
- ATAM $?_S$ TAM.
- Graham-Denning scheme $?_S$ HRU scheme.

- RT[ ] scheme $?_S$ HRU scheme.

Contents

HRU Scheme

- S = access matrix instances
- C = all command-sets, with each command:

```
command c($p_1$, $p_2$, ..., $p_n$)
  if $r_1 \in M[p_i,p_j]$ ? … ? $r_n \in M[p_k, p_l]$
    primitive op 1
    primitive op 2
    ...
```

  - Primitive op: create subject/object, destroy subject/object, enter/remove right.
- Q: (1) $r \in M[\sigma,\omega]$; (2) $r \notin M[\sigma,\omega]$

HRU Scheme (contd.)

- Safety problem: can a right appear where it does not exist in start-state?
  - Result: undecidable in general
- Import of result:
  - “Safety is undecidable in DAC”
  - “Shows limits of formal methods in security”
  - “HRU scheme is too expressive”

RT[ ] Scheme

- S = collection of assertions of two kinds:
  - A.r ? B (simple member)
  - A.r ? B.r_1 (simple inclusion)
- c = (G, H)
  - G: set of growth-restricted roles
  - H: set of shrink-restricted roles
- Q: (1) { B } ? A.r;

  (2) A.r ? { B };

  (3) A.r ? B.r_1

Result and Intuition

- RT[ ] scheme $?_S$ HRU scheme

- RT[ ] system:
  - Start with A.r being empty, and not growth-restricted.
  - Adding a single statement A.r ? B causes an unbounded number of queries of the form { B’ } ? A.r to become false.
- Any HRU system has to traverse “bad” state.
  - Only bounded number of queries can change from true to false (or vice versa) in single state-change.

Contents

Related Work

- Based on preservation of safety:
  - Sandhu (JCS, ’92)
  - Ammann, Lipton, Sandhu (JCS, ’96)
  - Sandhu, Ganta (CSFW, ’93)

- Not based on preservation of safety:
  - Bertino, Catania, Ferrari, Perlasca (TISSEC, ‘03)
  - Chander, Dean, Mitchell (CSFW, ’01)
  - Osborn, Sandhu, Munawer (TISSEC, ’00)

Summary

- A theory for comparing access control models based on expressive power.
- Validated with applications
  - ATAM, TAM relationship was an open problem
  - SDCO, ARBAC97 result contradicts existing assertion from literature
  - Results on HRU are first formal evidence of its limited expressive power