Lecture #6

- Schematic Protection Model
  - Safety question
- Expressive Power
  - HRU and SPM
- Multiparent create
  - ESPM
Key Question

• Characterize class of models for which safety is decidable
  – Existence: Take-Grant Protection Model is a member of such a class
  – Universality: In general, question undecidable, so for some models it is not decidable

• What is the dividing line?
Schematic Protection Model

• Type-based model
  – Protection type: entity label determining how control rights affect the entity
    • Set at creation and cannot be changed
  – Ticket: description of a single right over an entity
    • Entity has sets of tickets (called a domain)
    • Ticket is X/r, where X is entity and r right
  – Functions determine rights transfer
    • Link: are source, target “connected”?
    • Filter: is transfer of ticket authorized?
Link Predicate

• Idea: $\text{link}_i(X, Y)$ if $X$ can assert some control right over $Y$

• Conjunction of disjunction of:
  – $X/z \in \text{dom}(X)$
  – $X/z \in \text{dom}(Y)$
  – $Y/z \in \text{dom}(X)$
  – $Y/z \in \text{dom}(Y)$
  – true
Examples

• Take-Grant:
  \[ \text{link}(X, Y) = Y/g \in \text{dom}(X) \lor X/t \in \text{dom}(Y) \]

• Broadcast:
  \[ \text{link}(X, Y) = X/b \in \text{dom}(X) \]

• Pull:
  \[ \text{link}(X, Y) = Y/p \in \text{dom}(Y) \]
Filter Function

- Range is set of copyable tickets
  - Entity type, right
- Domain is subject pairs
- Copy a ticket $X/r:c$ from $dom(Y)$ to $dom(Z)$
  - $X/rc \in dom(Y)$
  - $link_i(Y, Z)$
  - $\tau(Y)/r:c \in f_i(\tau(Y), \tau(Z))$
- One filter function per link predicate
Example

• $f(\tau(Y), \tau(Z)) = T \times R$
  – Any ticket can be transferred (if other conditions met)

• $f(\tau(Y), \tau(Z)) = T \times RI$
  – Only tickets with inert rights can be transferred (if other conditions met)

• $f(\tau(Y), \tau(Z)) = \emptyset$
  – No tickets can be transferred
Example

- **Take-Grant Protection Model**
  - $TS = \{ \text{subjects} \}, \ TO = \{ \text{objects} \}$
  - $RC = \{ \text{tc}, \ gc \}, \ RI = \{ \text{rc}, \ wc \}$
  - $\text{link}(p, q) = p/t \in \text{dom}(q) \lor q/g \in \text{dom}(p)$
  - $f(\text{subject, subject}) = \{ \text{subject, object} \} \times \{ \text{tc, gc, rc, wc} \}$
Create Operation

- Must handle type, tickets of new entity
- Relation $cc(a, b)$ [cc for can-create]
  - Subject of type $a$ can create entity of type $b$
- Rule of acyclic creates:

```
  a   b
  |   |
  v   v
  c   d
```

```
  a   b
  |   |
  v   v
  c   d
```
Types

- \( cr(a, b) \): tickets created when subject of type \( a \) creates entity of type \( b \) [\( cr \) for \textit{create-rule}]
- \( B \) object: \( cr(a, b) \subseteq \{ b/r:c \in RI \} \)
  - \( A \) gets \( B/r:c \) iff \( b/r:c \in cr(a, b) \)
- \( B \) subject: \( cr(a, b) \) has two subsets
  - \( cr_P(a, b) \) added to \( A \), \( cr_C(a, b) \) added to \( B \)
  - \( A \) gets \( B/r:c \) if \( b/r:c \in cr_P(a, b) \)
  - \( B \) gets \( A/r:c \) if \( a/r:c \in cr_C(a, b) \)
Non-Distinct Types

\( cr(a, a) \): who gets what?

- \( self/r:c \) are tickets for creator
- \( a/r:c \) tickets for created

\[
ctr(a, a) = \{ a/r:c, self/r:c \mid r:c \in R \}
\]
Attenuating Create Rule

\[ \text{cr}(a, b) \text{ attenuating if:} \]

1. \( \text{cr}_C(a, b) \subseteq \text{cr}_P(a, b) \) and
2. \( a/r:c \in \text{cr}_P(a, b) \Rightarrow \text{self}/r:c \in \text{cr}_P(a, b) \)
Example: Owner-Based Policy

- Users can create files, creator can give itself any inert rights over file
  - \( cc = \{ (user, file) \} \)
  - \( cr(user, file) = \{ file/r:c \mid r \in RI \} \)
- Attenuating, as graph is acyclic, loop free

![Diagram](image-url)
Example: Take-Grant

- Say subjects create subjects (type \(s\)), objects (type \(o\)), but get only inert rights over latter
  - \(cc = \{(s, s), (s, o)\}\)
  - \(cr_c(a, b) = \emptyset\)
  - \(cr_p(s, s) = \{s/tc, s/gc, s/rc, s/wc\}\)
  - \(cr_p(s, o) = \{s/rc, s/wc\}\)

- Not attenuating, as no *self* tickets provided; *subject* creates *subject*
Safety Analysis

• Goal: identify types of policies with tractable safety analyses
• Approach: derive a state in which additional entries, rights do not affect the analysis; then analyze this state
  – Called a maximal state
Definitions

- System begins at initial state
- Authorized operation causes *legal transition*
- Sequence of legal transitions moves system into final state
  - This sequence is a *history*
  - Final state is *derivable* from history, initial state
More Definitions

• States represented by $h$
• Set of subjects $SUB^h$, entities $ENT^h$
• Link relation in context of state $h \ link^h$
• Dom relation in context of state $h \ dom^h$
\( \text{path}^h(X,Y) \)

- \( X, Y \) connected by one link or a sequence of links

- Formally, either of these hold:
  - for some \( i \), \( \text{link}^h_i(X,Y) \); or
  - there is a sequence of subjects \( X_0, \ldots, X_n \) such that \( \text{link}^h_i(X,X_0), \text{link}^h_i(X_n,Y) \), and for \( k = 1, \ldots, n \), \( \text{link}^h_i(X_{k-1},X_k) \)

- If multiple such paths, refer to \( \text{path}^h_j(X,Y) \)
Capacity $cap(path^h(X,Y))$

- Set of tickets that can flow over $path^h(X,Y)$
  - If $link_i^h(X,Y)$: set of tickets that can be copied over the link (i.e., $f_i(\tau(X), \tau(Y))$)
  - Otherwise, set of tickets that can be copied over *all* links in the sequence of links making up the $path^h(X,Y)$

- Note: all tickets (except those for the final link) *must* be copyable
Flow Function

- Idea: capture flow of tickets around a given state of the system
- Let there be $m$ path$^h$s between subjects $X$ and $Y$ in state $h$. Then flow function

$$flow^h: SUB^h \times SUB^h \to 2^{T \times R}$$

is:

$$flow^h(X,Y) = \bigcup_{i=1, \ldots, m} cap(path^h_i(X,Y))$$
Properties of Maximal State

• Maximizes flow between all pairs of subjects
  – State is called *
  – Ticket in flow*(X,Y) means there exists a sequence of operations that can copy the ticket from X to Y

• Questions
  – Is maximal state unique?
  – Does every system have one?
Formal Definition

- Definition: $g \leq_0 h$ holds iff for all $X, Y \in SUB^0$, $flow^g(X,Y) \subseteq flow^h(X,Y)$.
  - Note: if $g \leq_0 h$ and $h \leq_0 g$, then $g, h$ equivalent
  - Defines set of equivalence classes on set of derivable states

- Definition: for a given system, state $m$ is maximal iff $h \leq_0 m$ for every derivable state $h$

- Intuition: flow function contains all tickets that can be transferred from one subject to another
  - All maximal states in same equivalence class
Maximal States

• Lemma. Given arbitrary finite set of states $H$, there exists a derivable state $m$ such that for all $h \in H$, $h \leq_0 m$

• Outline of proof: induction
  – Basis: $H = \emptyset$; trivially true
  – Step: $|H'| = n + 1$, where $H' = G \cup \{h\}$. By IH, there is a $g \in G$ such that $x \leq_0 g$ for all $x \in G$. 
Outline of Proof

• $M$ interleaving histories of $g$, $h$ which:
  – Preserves relative order of transitions in $g$, $h$
  – Omits second create operation if duplicated

• $M$ ends up at state $m$

• If $\text{path}^g(X,Y)$ for $X, Y \in \text{SUB}^g$, $\text{path}^m(X,Y)$
  – So $g \leq_0 m$

• If $\text{path}^h(X,Y)$ for $X, Y \in \text{SUB}^h$, $\text{path}^m(X,Y)$
  – So $h \leq_0 m$

• Hence $m$ maximal state in $H'$
Answer to Second Question

• Theorem: every system has a maximal state *

• Outline of proof: $K$ is set of derivable states containing exactly one state from each equivalence class of derivable states
  – Consider $X, Y$ in $SUB^0$. Flow function’s range is $2^{|T \times R|}$, so can take at most $2^{|T \times R|}$ values. As there are $|SUB^0|^2$ pairs of subjects in $SUB^0$, at most $2^{|T \times R|} |SUB^0|^2$ distinct equivalence classes; so $K$ is finite

• Result follows from lemma
Safety Question

• In this model:
  Is there a derivable state with $X/r:c \in \text{dom}(A)$, or does there exist a subject $B$ with ticket $X/rc$ in the initial state in $\text{flow}^*(B,A)$?

• To answer: construct maximal state and test
  – Consider acyclic attenuating schemes; how do we construct maximal state?
Intuition

• Consider state $h$.
• State $u$ corresponds to $h$ but with minimal number of new entities created such that maximal state $m$ can be derived with no create operations
  – So if in history from $h$ to $m$, subject $X$ creates two entities of type $a$, in $u$ only one would be created; surrogate for both
• $m$ can be derived from $u$ in polynomial time, so if $u$ can be created by adding a finite number of subjects to $h$, safety question decidable.
Fully Unfolded State

• State $u$ derived from state 0 as follows:
  – delete all loops in $cc$; new relation $cc'$
  – mark all subjects as folded
  – while any $X \in SUB^0$ is folded
    • mark it unfolded
    • if $X$ can create entity $Y$ of type $y$, it does so (call this the $y$-surrogate of $X$); if entity $Y \in SUB^g$, mark it folded
  – if any subject in state $h$ can create an entity of its own type, do so

• Now in state $u$
Termination

• First loop terminates as $SUB^0$ finite
• Second loop terminates:
  – Each subject in $SUB^0$ can create at most $|TS|$ children, and $|TS|$ is finite
  – Each folded subject in $|SUB^i|$ can create at most $|TS|$ – $i$ children
  – When $i = |TS|$, subject cannot create more children; thus, folded is finite
  – Each loop removes one element
• Third loop terminates as $SUB^h$ is finite
Surrogate

- Intuition: surrogate collapses multiple subjects of same type into single subject that acts for all of them
- Definition: given initial state $0$, for every derivable state $h$ define surrogate function $\sigma: \text{ENT}^h \rightarrow \text{ENT}^h$ by:
  - if $X$ in $\text{ENT}^0$, then $\sigma(X) = X$
  - if $Y$ creates $X$ and $\tau(Y) = \tau(X)$, then $\sigma(X) = \sigma(Y)$
  - if $Y$ creates $X$ and $\tau(Y) \neq \tau(X)$, then $\sigma(X) = \tau(Y)$- surrogate of $\sigma(Y)$
Implications

- $\tau(\sigma(X)) = \tau(X)$
- If $\tau(X) = \tau(Y)$, then $\sigma(X) = \sigma(Y)$
- If $\tau(X) \neq \tau(Y)$, then
  - $\sigma(X)$ creates $\sigma(Y)$ in the construction of $u$
  - $\sigma(X)$ creates entities $X'$ of type $\tau(X) = \tau(\sigma(X))$
- From these, for a system with an acyclic attenuating scheme, if $X$ creates $Y$, then tickets that would be introduced by pretending that $\sigma(X)$ creates $\sigma(Y)$ are in $\text{dom}^u(\sigma(X))$ and $\text{dom}^u(\sigma(Y))$
Deriving Maximal State

• Idea
  – Reorder operations so that all creates come first and replace history with equivalent one using surrogates
  – Show maximal state of new history is also that of original history
  – Show maximal state can be derived from initial state
Reordering

- $H$ legal history deriving state $h$ from state 0
- Order operations: first create, then demand, then copy operations
- Build new history $G$ from $H$ as follows:
  - Delete all creates
  - “$X$ demands $Y/r:c$” becomes “$\sigma(X)$ demands $\sigma(Y)/r:c$”
  - “$Y$ copies $X/r:c$ from $Y$” becomes “$\sigma(Y)$ copies $\sigma(X)/r:c$ from $\sigma(Y)$
Tickets in Parallel

• Theorem
  – All transitions in $G$ legal; if $X/r:c \in \text{dom}^h(Y)$, then $\sigma(X)/r:c \in \text{dom}^h(\sigma(Y))$

• Outline of proof: induct on number of copy operations in $H$
Basis

- $H$ has create, demand only; so $G$ has demand only. $s$ preserves type, so by construction every demand operation in $G$ legal.

- 3 ways for $X/r:c$ to be in $dom^h(Y)$:
  - $X/r:c \in dom^0(Y)$ means $X, Y \in ENT^0$, so trivially $\sigma(X)/r:c \in dom^g(\sigma(Y))$ holds
  - A create added $X/r:c \in dom^h(Y)$: previous lemma says $\sigma(X)/r:c \in dom^g(\sigma(Y))$ holds
  - A demand added $X/r:c \in dom^h(Y)$: corresponding demand operation in $G$ gives $\sigma(X)/r:c \in dom^g(\sigma(Y))$
Hypothesis

- Claim holds for all histories with $k$ copy operations
- History $H$ has $k+1$ copy operations
  - $H'$ initial sequence of $H$ composed of $k$ copy operations
  - $h'$ state derived from $H'$
Step

• $G'$ sequence of modified operations corresponding to $H'$; $g'$ derived state
  – $G'$ legal history by hypothesis

• Final operation is “$Z$ copied $X/r:c$ from $Y$”
  – So $h, h'$ differ by at most $X/r:c \in \text{dom}^h(Z)$
  – Construction of $G$ means final operation is
    $\sigma(X)/r:c \in \text{dom}^g(\sigma(Y))$

• Proves second part of claim
Step

- \( H' \) legal, so for \( H \) to be legal, we have:
  1. \( X/rc \in \text{dom}^{h'}(Y) \)
  2. \( \text{link}^{h'}_i(Y, Z) \)
  3. \( \tau(X/r:c) \in f_i(\tau(Y), \tau(Z)) \)

- By IH, 1, 2, as \( X/r:c \in \text{dom}^{h'}(Y) \),
  \( \sigma(X)/r:c \in \text{dom}^{g'}(\sigma(Y)) \) and \( \text{link}^{g'}_i(\sigma(Y), \sigma(Z)) \)

- As \( \sigma \) preserves type, IH and 3 imply
  \( \tau(\sigma(X)/r:c) \in f_i(\tau((\sigma(Y)), \tau(\sigma(Z))) \)

- IH says \( G' \) legal, so \( G \) is legal
Corollary

• If $link_i^h(X, Y)$, then $link_i^g(\sigma(X), \sigma(Y))$