Lecture #4

- Conspiracy in the Take-Grant Protection Model
- *de facto* rules (information flow)
- Knowing in a combined graph
- Basics of Schematic Protection Model
Conspiracy

- Minimum number of actors to generate a witness for \( \text{can}\cdot\text{share}(\alpha, x, y, G_0) \)
  - Actor is defined as \( x \) such that \( x \) initiates \( Q_i \)
- Access set describes the “reach” of a subject
- Deletion set is set of vertices that cannot be involved in a transfer of rights
- Build \textit{conspiracy graph} to capture how rights flow, and derive actors from it
Access Set

- **Access set** $A(y)$ with focus $y$: set of vertices:
  - $\{ y \}$
  - $\{ x \mid y \text{ initially spans to } x \}$
  - $\{ x \mid y \text{ terminally spans to } x \}$

- Idea is that focus can give rights to, or acquire rights from, a vertex in this set.
Example

- $A(x) = \{ x, a \}$
- $A(b) = \{ b, a \}$
- $A(c) = \{ c, b, d \}$
- $A(d) = \{ d \}$
- $A(e) = \{ e, d, i, j \}$
- $A(h) = \{ h, f, i \}$
- $A(f) = \{ f, y \}$
- $A(y) = \{ y \}$
$tg$-sink

- $x_0$, only incoming $t$ edge
- $x_i$, two incoming incident edges, both labeled $t$ or both labeled $g$
- $x_n$, only incoming $g$ edge
Necessity

• Lower bound on number of conspirators
  – Rights can be transmitted to any vertex in the access set
  – Rights can be “passed along” through the overlap of access sets, unless common vertex cannot initiate rule (tg-sink)
  – If only common vertex is tg-sink, must aid in transfer
Necessity Theorem

- Let $can\cdot share(\alpha, p, q, G)$ hold, and define $G_0$ to be $G-\{q\}$. Let $k$ be the number of access sets in a minimal cover of $G_0$, and let $l$ be the number of $tg$-sinks. Then $k + l$ initiators are necessary to witness $can\cdot share(\alpha, p, q, G)$. 
Deletion Set

- Deletion set $\delta(y, y')$: contains those vertices in $A(y) \cap A(y')$ such that:
  - $y$ initially spans to $z$ and $y'$ terminally spans to $z$;
  - $y$ terminally spans to $z$ and $y'$ initially spans to $z$;
  - $z = y$
  - $z = y'$

- Idea is that rights can be transferred between $y$ and $y'$ if this set non-empty
Example

- $\delta(x, b) = \{a\}$
- $\delta(b, c) = \{b\}$
- $\delta(c, d) = \{d\}$
- $\delta(c, e) = \{d\}$
- $\delta(d, e) = \{d\}$
- $\delta(y, f) = \{y\}$
- $\delta(h, f) = \{f\}$
- $\delta(e, h) = \emptyset$
Sufficiency

• Consider \( A(x_i) \cap A(x_{i+1}) = \{ y \} \)
  – If edges incoming to \( y \) are both \( t \) or both \( g \), \( y \) must act
  – If edges incoming to \( y \) are \( t \) and \( g \), it’s a bridge and \( y \) need not act

• So, in first case, need one additional operation initiated by \( y \)

• Note: \( y \) is a \( tg \)-sink in these cases
Conspiracy Graph

- Abstracted graph $H$ from $G_0$:
  - Each subject $x \in G_0$ corresponds to a vertex $h(x) \in H$
  - If $\delta(x, y) \neq \emptyset$, there is an edge between $h(x)$ and $h(y)$ in $H$

- Idea is that if $h(x), h(y)$ are connected in $H$, then rights can be transferred between $x$ and $y$ in $G_0$
Example

\[ h(x) \rightarrow h(y) \rightarrow h(f) \rightarrow h(h) \]

\[ h(b) \rightarrow h(g) \rightarrow h(d) \rightarrow h(e) \]

\[ h(c) \rightarrow h(t) \rightarrow h(g) \rightarrow h(q) \]

January 13, 2011  ECS 235B Winter Quarter 2011  Slide #4-12
Sharing

- $I(x)$: $h(x)$, all vertices $h(y)$ such that $y$ initially spans to $x$
- $T(x)$: $h(x)$, all vertices $h(y)$ such that $y$ terminally spans to $x$
- Theorem: $\text{can\cdot share}(\alpha, x, y, G_0)$ iff there exists a path from some $h(p)$ in $I(x)$ to some $h(q)$ in $T(y)$
  - Idea: path exists if access sets overlap and rights can be transferred between endpoints
  - Note $tg$-sinks correspond to singleton access sets with foci that must act (idea of deletion sets)
Counting Conspirators

- Theorem: if there are $l$ vertices on shortest path between $h(p)$, $h(q)$ in above theorem, $l$ conspirators necessary and sufficient to witness
  - Follows immediately from previous two theorems, definitions
Example: Conspirators

- $I(x) = \{ h(x) \}$, $T(z) = \{ h(e) \}$
- Path between $h(x)$, $h(e)$ so can • share$(r, x, z, G_0)$
- Shortest path between $h(x)$, $h(e)$ has 4 vertices
  \[ \Rightarrow \text{Conspirators are } e, c, b, x \]
Example: Witness

- e grants (r to z) to d
- c takes (r to z) from d
- c grants (r to z) to b
- b grants (r to z) to a
- x takes (r to z) from a
**de facto** Rules

- These deal with information flow
- Not graph rewriting rules
  - Add no edges
  - Instead, represent flows by “implicit” edges, shown as:

```
  x  r  y
```

Rules

spy

find

post

pass
Example

\[ u \text{ posts through } s \text{ to } p \]
\[ v \text{ passes from } w \text{ to } u \]
\[ w \text{ spies through } x \text{ to } q \]
\[ u \text{ spies through } w \text{ to } q \]
\[ p \text{ spies through } u \text{ to } q \]
**can•know**

Definition:

- \( can•know(x, y, G_0) \) if, and only if, there is a sequence of protection graphs \( G_0, \ldots, G_n \) such that \( G_0 \vdash^* G_n \) using *de jure* or *de facto* rules and in \( G_n \) there is an edge from \( x \) to \( y \) labeled \( r \).
Example

y creates (rw to new) z
x takes (r to z) from y
y passes to x through z
x takes (w to z) from y
y posts to x through z
Combined Transfers

The subject can acquire $\alpha$ rights over the last object.

The subject can acquire $r$ rights over the last object.
Combined Transfers

The subject can acquire $g$ rights over the last object.

The subject can acquire $w$ rights over the last object.
Combined Transfers

Just as rights can be transferred over a bridge, information can flow over a connection
Theorem

$can\cdot know(p, q, G_0)$ holds if and only if:

(a) $can\cdot share(r, p, q, G_0)$ holds, or

(b) there is a sequence of subjects $u_1, \ldots, u_n$ such that all of the following are true:

(i) $p = u_1$ or $u_1$ rw-initially spans to $p$;

(ii) $q = u_n$ or $u_n$ rw-terminally spans to $q$; and

(iii) for all $i, 1 \leq i < n$, there is an $rwtg$-path between $u_i$ and $u_{i+1}$ with associated word a bridge or connection.
can•know\((p, q, G_0)\) holds:

- take \(n = 2\), \(u_1 = x\), and \(u_2 = y\)
  - \(p = u_1\) or \(u_1\) rw-initially spans to \(p\);
  - \(q = u_2\) or \(u_2\) rw-terminally spans to \(q\); and
  - there is an \(rwtg\)-path between \(u_1\) and \(u_2\) with associated word a bridge or connection
- \(u_1, u_2\) connected with a \(t\) edge
can\-\*\-share\((r, v, z, G_0)\) is false
• no initial span between \(v\) and any subject

can\-\*\-know\((v, z, G_0)\) is true
• \(u_1 = w, u_2 = x\)
• \(u_1\) \(rw\)-initially spans to \(y\)
• \(u_2\) \(rw\)-terminally spans to \(z\)
• there is a connection between \(u_1\) and \(u_2\)
**Final Example Witness**

\[ v \xrightarrow{w} w \xrightarrow{r} x \xrightarrow{r} y \xrightarrow{r} z \]

- \text{\textbf{x}} \text{ takes (r to z) from y}
- \text{\textbf{w}} \text{ spies on z through x}
- \text{\textbf{w}} \text{ passes from z to v}
- \text{\textbf{w}} \text{ passes from x to v}
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:
  \[ link(X, Y) = Y/g \in dom(X) \lor X/t \in dom(Y) \]

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

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

- Range is set of copyable tickets
  - Entity type, right
- Domain is subject pairs
- Copy a ticket $X/r:c$ from $\text{dom}(Y)$ to $\text{dom}(Z)$
  - $X/rc \in \text{dom}(Y)$
  - $\text{link}_i(Y, Z)$
  - $\tau(Y)/r:c \in f_i(\tau(Y), \tau(Z))$
- 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
  – $TS = \{ \text{subjects} \}$, $TO = \{ \text{objects} \}$
  – $RC = \{ tc, gc \}$, $RI = \{ rc, wc \}$
  – $\text{link}(p, q) = p/t \in \text{dom}(q) \lor q/g \in \text{dom}(p)$
  – $f(\text{subject}, \text{subject}) = \{ \text{subject}, \text{object} \} \times \{ tc, gc, rc, wc \}$
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:
Types

- \( cr(a, b) \): tickets created when subject of type \( a \) creates entity of type \( b \) [\( cr \) for 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

\[ cr(a, a): \text{who gets what?} \]

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

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

$cr(a, b)$ attenuating if:

1. $cr_C(a, b) \subseteq cr_P(a, b)$ and
2. $a/r:c \in cr_P(a, b) \Rightarrow self/r:c \in cr_P(a, b)$
Example: Owner-Based Policy

- Users can create files, creator can give itself any inert rights over file
  - $cc = \{ (\text{user}, \text{file}) \}$
  - $cr(\text{user}, \text{file}) = \{ \text{file}/r:c \mid r \in RI \}$

- Attenuating, as graph is acyclic, loop free
Example: Take-Grant

- Say subjects create subjects (type \( s \)), objects (type \( o \)), but get only inert rights over latter
  - \( cc = \{(s,s),(s,o)\} \)
  - \( cr_c(a,b) = \emptyset \)
  - \( cr_p(s,s) = \{s/tc,s/gc,s/rc,s/wc\} \)
  - \( cr_p(s,o) = \{o/rc,o/wc\} \)
- Not attenuating, as no \textit{self} tickets provided; \textit{subject} creates \textit{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 $\text{SUB}^h$, entities $\text{ENT}^h$
- Link relation in context of state $h \text{ link}^h$
- Dom relation in context of state $h \text{ dom}^h$
path^h(X,Y)

- X, Y connected by one link or a sequence of links
- Formally, either of these hold:
  - for some i, link^i_h(X, Y); or
  - there is a sequence of subjects X_0, ..., X_n such that link^i_h(X, X_0), link^i_h(X_n, Y), and for k = 1, ..., n, link^i_h(X_{k-1}, X_k)
- If multiple such paths, refer to path^j_h(X, Y)