April 7: Safety Question

- Protection State Transitions
  - Commands
  - Conditional Commands
- Special Rights
  - Principle of Attenuation of Privilege
- Harrison-Ruzzo-Ullman result
  - Corollaries
General Case

- Answer: no

- Sketch of proof:
  Reduce halting problem to safety problem
  Turing Machine review:
  - Infinite tape in one direction
  - States $K$, symbols $M$; distinguished blank $b$
  - Transition function $\delta(k, m) = (k', m', L)$ means in state $k$, symbol $m$ on tape location replaced by symbol $m'$, head moves to left one square, and enters state $k'$
  - Halting state is $q_f$; TM halts when it enters this state
## Mapping

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>( \cdots )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>head</td>
</tr>
</tbody>
</table>

Current state is \( k \)

<table>
<thead>
<tr>
<th></th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>( s_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>A</td>
<td>own</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( s_2 )</td>
<td>B</td>
<td>own</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( s_3 )</td>
<td>C</td>
<td>( k )</td>
<td>own</td>
<td></td>
</tr>
<tr>
<td>( s_4 )</td>
<td></td>
<td></td>
<td>D</td>
<td>end</td>
</tr>
</tbody>
</table>
After $\delta(k, C) = (k_1, X, R)$ where $k$ is the current state and $k_1$ the next state
\( \delta(k, C) = (k_1, X, R) \) at intermediate becomes

\[
\text{command } c_{k, C}(s_3, s_4)
\]

\[
\text{if } \text{own in } A[s_3, s_4] \text{ and } k \text{ in } A[s_3, s_3] \quad \text{and } C \text{ in } A[s_3, s_3]
\]

\[
\text{then}
\]

\[
\text{delete } k \text{ from } A[s_3, s_3];
\]

\[
\text{delete } C \text{ from } A[s_3, s_3];
\]

\[
\text{enter } X \text{ into } A[s_3, s_3];
\]

\[
\text{enter } k_1 \text{ into } A[s_4, s_4];
\]

\text{end}
After $\delta(k_1, D) = (k_2, Y, R)$ where $k_1$ is the current state and $k_2$ the next state.
Command Mapping

\[ \delta(k_1, D) = (k_2, Y, R) \] at end becomes

\textbf{command} c\text{rightmost}_{k,c}(s_4, s_5)

\textbf{if} \text{ end in } A[s_4, s_4] \text{ and } k_1 \text{ in } A[s_4, s_4]

\text{and} \ D \text{ in } A[s_4, s_4]

\textbf{then}

\text{delete} \ \text{end from} \ A[s_4, s_4];

\text{delete} \ k_1 \ \text{from} \ A[s_4, s_4];

\text{delete} \ D \ \text{from} \ A[s_4, s_4];

\text{enter} \ Y \ \text{into} \ A[s_4, s_4];

\text{create subject} \ s_5;

\text{enter} \ own \ \text{into} \ A[s_4, s_5];

\text{enter} \ end \ \text{into} \ A[s_5, s_5];

\text{enter} \ k_2 \ \text{into} \ A[s_5, s_5];

\textbf{end}
Rest of Proof

• Protection system exactly simulates a TM
  – Exactly 1 end right in ACM
  – 1 right in entries corresponds to state
  – Thus, at most 1 applicable command

• If TM enters state $q_f$, then right has leaked

• If safety question decidable, then represent TM as above and determine if $q_f$ leaks
  – Implies halting problem decidable

• Conclusion: safety question undecidable
Other Results

- Set of unsafe systems is recursively enumerable
- Delete `create` primitive; then safety question is complete in \( P\)-SPACE
- Delete `destroy`, `delete` primitives; then safety question is undecidable
  - Systems are monotonic
- Safety question for biconditional protection systems is decidable
- Safety question for monoconditional, monotonic protection systems is decidable
- Safety question for monoconditional protection systems with `create`, `enter`, `delete` (and no `destroy`) is decidable.
Take-Grant Protection Model

- A specific (not generic) system
  - Set of rules for state transitions
- Safety decidable, and in time linear with the size of the system
- Goal: find conditions under which rights can be transferred from one entity to another in the system
System

- objects (files, …)
- subjects (users, processes, …)
- don't care (either a subject or an object)

\[ G \vdash_x G' \] apply a rewriting rule \( x \) (witness) to \( G \) to get \( G' \)

\[ G \vdash^* G' \] apply a sequence of rewriting rules (witness) to \( G \) to get \( G' \)

\[ R = \{ t, g, r, w, \ldots \} \] set of rights
Rules

take

\[ t \alpha \]

grant

\[ g \alpha \]
More Rules

create

\[ \alpha \rightarrow \times \]

\[ \vdash \alpha \rightarrow \times \]

remove

\[ \alpha \rightarrow \times \]

\[ \vdash \alpha - \beta \rightarrow \times \]

These four rules are called the *de jure* rules
Symmetry

1. $x$ creates $(tg \text{ to new}) \, v$
2. $z$ takes $(g \text{ to } v)$ from $x$
3. $z$ grants $(\alpha \text{ to } y)$ to $v$
4. $x$ takes $(\alpha \text{ to } y)$ from $v$

Similar result for grant
Islands

• $tg$-path: path of distinct vertices connected by edges labeled $t$ or $g$
  – Call them “$tg$-connected”

• island: maximal $tg$-connected subject-only subgraph
  – Any right one vertex has can be shared with any other vertex
Initial, Terminal Spans

• *initial span* from \( x \) to \( y \)
  
  – \( x \) subject
  
  – \( tg \)-path between \( x, y \) with word in \{ \( t^*g \) \} \( \cup \) \{ \( \nu \) \}
  
  – Means \( x \) can give rights it has to \( y \)

• *terminal span* from \( x \) to \( y \)
  
  – \( x \) subject
  
  – \( tg \)-path between \( x, y \) with word in \{ \( t^* \) \} \( \cup \) \{ \( \nu \) \}
  
  – Means \( x \) can acquire any rights \( y \) has
• bridge: \(tg\)-path between subjects \(x, y\), with associated word in

\[
\left\{ \overrightarrow{t^*}, \overleftarrow{t^*}, \overrightarrow{t^*g} \overleftarrow{t^*}, \overrightarrow{t^*g} \overrightarrow{t^*} \right\}
\]

– rights can be transferred between the two endpoints

– *not* an island as intermediate vertices are objects
Example

- islands: \( \{ p, u \} \) \( \{ w \} \) \( \{ y, s' \} \)
- bridges: \( u, v, w; w, x, y \)
- initial span: \( p \) (associated word \( v \))
- terminal span: \( s' \)’s (associated word \( t' \))
can•share Predicate

Definition:

• \textit{can•share}(r, 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 only \textit{de jure} rules and in $G_n$ there is an edge from $x$ to $y$ labeled $r$. 
**can•share** Theorem

- **can•share**\((r, x, y, G_0)\) if, and only if, there is an edge from \(x\) to \(y\) labeled \(r\) in \(G_0\), or the following hold simultaneously:
  - There is an \(s\) in \(G_0\) with an \(s\)-to-\(y\) edge labeled \(r\)
  - There is a subject \(x' = x\) or initially spans to \(x\)
  - There is a subject \(s' = s\) or terminally spans to \(s\)
  - There are islands \(I_1, \ldots, I_k\) connected by bridges, and \(x'\) in \(I_1\) and \(s'\) in \(I_k\)
Outline of Proof

• s has r rights over y
• s' acquires r rights over y from s
  – Definition of terminal span
• x' acquires r rights over y from s'
  – Repeated application of sharing among vertices in islands, passing rights along bridges
• x' gives r rights over y to x
  – Definition of initial span
Example Interpretation

• ACM is generic
  – Can be applied in any situation

• Take-Grant has specific rules, rights
  – Can be applied in situations matching rules, rights

• Question: what states can evolve from a system that is modeled using the Take-Grant Model?
Take-Grant Generated Systems

- Theorem: $G_0$ protection graph with 1 vertex, no edges; $R$ set of rights. Then $G_0 \vdash * G$ iff:
  - $G$ finite directed graph consisting of subjects, objects, edges
  - Edges labeled from nonempty subsets of $R$
  - At least one vertex in $G$ has no incoming edges
Outline of Proof

⇒: By construction; $G$ final graph in theorem
- Let $x_1, \ldots, x_n$ be subjects in $G$
- Let $x_1$ have no incoming edges

• Now construct $G'$ as follows:
  1. Do "$x_1$ creates ($\alpha \cup \{ g \}$ to) new subject $x_i"$
  2. For all $(x_i, x_j)$ where $x_i$ has a rights over $x_j$, do
     "$x_1$ grants ($\alpha$ to $x_j$) to $x_i"$
  3. Let $\beta$ be rights $x_i$ has over $x_j$ in $G$. Do
     "$x_1$ removes (($\alpha \cup \{ g \}$ – $\beta$ to) $x_j"$

• Now $G'$ is desired $G$
Outline of Proof

⇐: Let $v$ be initial subject, and $G_0 \vdash^* G$

• Inspection of rules gives:
  – $G$ is finite
  – $G$ is a directed graph
  – Subjects and objects only
  – All edges labeled with nonempty subsets of $R$

• Limits of rules:
  – None allow vertices to be deleted so $v$ in $G$
  – None add incoming edges to vertices without incoming edges, so $v$ has no incoming edges
Example: Shared Buffer

- Goal: \( p, q \) to communicate through shared buffer \( b \) controlled by trusted entity \( s \)
  1. \( s \) creates (\( \{r, w\} \) to new object) \( b \)
  2. \( s \) grants (\( \{r, w\} \) to \( b \)) to \( p \)
  3. \( s \) grants (\( \{r, w\} \) to \( b \)) to \( q \)
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 $\text{dom}(Y)$ 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 function
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
  
  – $TS = \{ \text{subjects} \}$, $TO = \{ \text{objects} \}$
  
  – $RC = \{ tc, gc \}$, $RI = \{ rc, wc \}$
  
  – $\text{link}(p, q) = p/t \in \text{dom}(q) \vee q/g \in \text{dom}(p)$
  
  – $f(\text{subject, subject}) = \{ \text{subject, object} \} \times \{ tc, gc, rc, wc \}$