- Exam will move to earlier date
- All notes submitted so far are OK

**Access Control**

- ACM (Access Control Matrix)
  - capabilities
  - ACLs
- Bell-LaPadula/Biba/Lattice-Based ACs
- Role-Based Access Control (RBAC)

**Homework Grading System**

- Apps only communicate via messages
- OS tags each message with sender
- Only student can see own grade
- Can only submit homework once
**ACM - Access Control Matrix**

\[ A_{d,0} = \{ r \mid d \text{ has } r \text{ access to } o \}\]

<table>
<thead>
<tr>
<th>A</th>
<th>HW Queue</th>
<th>Student 1 Grade</th>
<th>Student 2 Grade</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor</td>
<td>dequeue</td>
<td>owner</td>
<td>read</td>
<td>read</td>
</tr>
<tr>
<td></td>
<td>enqueue</td>
<td></td>
<td>write</td>
<td>write</td>
</tr>
<tr>
<td>Student 1</td>
<td>enqueue</td>
<td></td>
<td>read</td>
<td></td>
</tr>
<tr>
<td>Student 2</td>
<td>enqueue</td>
<td></td>
<td>read</td>
<td>read</td>
</tr>
</tbody>
</table>

Owner ∈ A_{d,0} the d can grant d′ r on o.
Control ∈ A_{d1,d2} then d1 can rename r for A_{d1,0}
copy bit ⇒ can delete your privileges.

An access control system consists of
- access control matrix
- set of rights
- set of commands

**Command** \((d_1, \ldots, d_n, r_1, \ldots, r_m, o_1, \ldots, o_k)\)
\[ \text{if } (r_i \in A_{d_i,0} \text{ and } r_j \in A_{d_j,0,2} \text{ and } \ldots) \]

\(\text{Op}_1 : \{ \text{operations are adding and removing from ACM} \}
\)

**grant** \((d, d, r, o)\)
\[ \text{if } (\text{owner} \in A_{d,0}) \]
\[ A_{d,0} U = \{ r \} \]

**Question:** Is there a sequence of commands that can result in \( r \in A_{d,0} \)?

Theorem: Harrison - Ruzzo - Ullman
This is undecidable.
Implementing ACMs

- Capability Lists
  - store by row
  - with each domain, store a list of its access rights
  - easy to look up domain capabilities
  - file handles

- Access Control Lists
  - store by column
  - list everyone that can access an object with that object
  - easy to see who can access something
  - file permissions
  - Windows NTFS, AndrewFS, POSIX Linux
  - Positive/Negative order?
  - rules should be unordered

Bell-LaPadula (Multilevel Security)

If o is labeled with $S$, the d can only access o if $d$ is labeled with $S' \geq S$

Secrecy: \begin{align*}
  \text{Top Secret} & \quad \text{Compartment: JFK} \\
  \text{Secret} & \quad \text{Area 51} \\
  \text{Confidential} & \\
  \text{Public} & 
\end{align*}

If o is labeled with $C = \{C_0, \ldots, C_3\}$ then a domain must have access to all compartments in C

A "label" is a pair $\lambda = (s, C)$ \{C, s set of compartments\}

we define $(s_1, C_1) \leq (s_2, C_2)$

iff $s_1 \leq s_2$

$C_1 \subseteq C_2$
LUB \((l_1, l_2) = l\) s.t.

1. \(l_1 \leq l\)
2. \(l_2 \leq l\)
3. \(\forall l', l_1 \leq l' \text{ and } l_2 \leq l' \text{ and } l \leq l'\)

What about writes?

If \(d\) has label \(l\) and \(o\) has label \(l'\),

- \(d\) can write \(o\) if \(l \leq l'\)
- \(d\) can read \(o\) if \(l' \leq l\)