Outline for October 7, 2014

Reading: *text*, §3.1–3.2; paper [TL13]

1. Primitive operations
   a. **enter** r **into** $A[s, o]$
   b. **delete** r **from** $A[s, o]$
   c. **create subject** s (note that $\forall x[A[s', x] = A[x, s'] = \varnothing]$)
   d. **create object** o (note that $\forall x[A[x, o'] = \varnothing]$)
   e. **destroy subject** s
   f. **destroy object** o
2. Commands and examples
   a. Regular command: *create·file*
   b. Mono-operational command: *make·owner*
   c. Conditional command: *grant·rights*
   d. Biconditional command: *grant·read·if·r·and·c*
   e. Doing "or" of 2 conditions: *grant·read·if·r·or·c*
3. Miscellaneous points
   a. Copy flag and right
   b. Own as a distinguished right
   c. Principle of attenuation of privilege
4. What is the safety question?
   a. An unauthorized state is one in which a generic right r could be leaked into an entry in the ACM that did not previously contain r. An initial state is safe for r if it cannot lead to a state in which r could be leaked.
   b. Question: in a given arbitrary protection system, is safety decidable?
5. Mono-operational case: there is an algorithm that decides whether a given mono-operational system and initial state is safe for a given generic right.
6. General case: It is undecidable whether a given state of a given protection system is safe for a given generic right.
   a. Approach: represent Turing machine tape as access control matrix, transitions as commands
   b. Reduce halting problem to it
7. Related results
   a. The set of unsafe systems is recursively enumerable
   b. Monotonicity: no delete or destroy primitive operations
   c. The safety question for biconditional monotonic protection systems is undecidable.
   d. The safety question for monoconditional monotonic protection systems is decidable.
   e. The safety question for monoconditional protection systems without the destroy primitive operation is decidable.