Notes for February 4, 1998

1. Greetings and felicitations!
   a. **Reading**: Pfleeger, pp. 247–249, 277–280, 304–305
2. Puzzle
   a. Point is that sometimes non-technical solutions are the most effective
3. MULTICS ring mechanism
   a. MULTICS rings: used for both data and procedures; rights are REWA
   b. (b1, b2) access bracket - can access freely; (b3, b4) call bracket - can call segment through gate; so if a's access bracket is (32,35) and its call bracket is (36,39), then *assuming permission mode (REWA) allows access*, a procedure in:
   rings 0-31: can access a, but ring-crossing fault occurs
   rings 32-35: can access a, no ring-crossing fault
   rings 36-39: can access a, provided a valid gate is used as an entry point
   rings 40-63: cannot access a
   c. If the procedure is accessing a data segment d, no call bracket allowed; given the above, *assuming permission mode (REWA) allows access*, a procedure in:
   rings 0-32: can access d
   rings 33-35: can access d, but cannot write to it (W or A)
   rings 36-63: cannot access d
4. Capabilities
   a. Capability-based addressing: show picture of accessing object
   b. Show process limiting access by not inheriting all parent's capabilities
   c. Revocation: use of a global descriptor table

[*ended here*]

5. Lock and Key
   a. Associate with each object a lock; associate with each process that has access to object a key (it's a cross between ACLs and C-Lists)
   b. Example: use crypto (Gifford). X object enciphered with key K. Associate an opener R with X. Then:
   OR-Access: K can be recovered with any D_i in a list of n deciphering transformations, so $R = (E_1(K), E_2(K), ..., E_n(K))$ and any process with access to any of the D_i's can access the file
   AND-Access: need all n deciphering functions to get K: $R = E_1(E_2(...E_n(K)...))$
6. Mandatory vs. Discretionary;
   a. security levels
   b. categories
7. Bell-LaPadula Model
   a. Simple Security Property: no reads up

b. Star Property: no writes down

c. Discretionary Security Property: if mandatory controls say it's okay, check discretionary controls.

d. Basic Security Theorem: A system is secure if its initial state is secure and no action violates the above rules.

8. ORCON (Originator Controlled; Graubert)

a. Document/information can be passed on with approval of originator; real world justification is that originator of document trusts recipients not to release documents which they should not.

b. Untrusted subject *x* marks object *O* ORCON on behalf of organization *X* and indicates it is releasable to subjects acting on behalf of organization *Y.*
not releasable to subjects acting on behalf of other organizations without *X*'s permission
any copies made have the same restriction

c. DAC: can't do this as the restriction would not copy over (*y* reads *O* into *C*, puts its own ACL on *C*)

d. MAC: separate category with*O*, *x*, *y. y* wants to read *O*, copy to *C*; MAC means *C* has same category as *O*, *x*, *y*, so can't give *z* access to *C*.
Say a new organization *w* wants to provide data in *B* to *y* but not to be shared with *x* or *z*. Can't use *O*'s category. Hence you get explosion of categories.
Real world parallel: individuals are "briefed" into a category and those represent a formal "need to know" policy that is standard across the entity; ORCON has no central clearinghouse to categorize data; originator makes rules.

9. Solution?

a. owner of object can't change ACL's relationship with object (MAC characteristic)

b. on copy, ACL is copied as well (MAC characteristic)

c. access control restrictions can be tailored on a subject/object basis (DAC characteristic)