Outline for February 18, 1999

1. Greetings and felicitations!
   a. Can people make a make-up class on Monday at 10:30AM?
2. Representing access control
   a. ACM
   b. ACLs – columns: (subject, rights)
   c. C-Lists – rows: (object, rights); use ticket analogy
3. Capabilities Implementation
   a. Tagged Architecture: extra bits setting word so it can only be altered in privileged mode
   b. Cryptography (for a network or when no tags available): digitally sign capability with OS key
   c. Protection: keep capability in system area, OS manipulates them
   d. Copy right: can capabilities be inherited or copied? depends ...
4. Revocation
   a. ACL: just delete entry giving subject access to object
   b. C-Lists: can track down all capabilities; better to use indirection and aliasing through a Global Object Table
5. Discretionary AC Attacks: Trojan Horse
   a. overt - example edit a file
   b. covert - example delete all files
   c. a type of malicious logic (discuss this)
6. Approaches
   a. Mandatory Access Control; works between compartments, but not within a single compartment
   b. Limited Protection Domain: easiest with C-list; if not, can be widened using TH, especially if ACLs are used and child has privileges of initiator
   c. Name-checking subsystem; catches accesses not in pattern (startup, .asm, .obj)
7. Reference Monitor
   a. Controls access to a resource
   b. Verifiable: KISS Principlee
   c. Complete: should only be able to get to resource through the monitor
   d. Tamperproof: can't be changed without authorization
8. 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
9. 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: $R = E_1(E_2(...E_n(K)...))$