manipulate
E1. The system must maintain these lists and must ensure only those TPs manipulate those CDIs
E2: The system must maintain a list of User IDs, TP, and CDIs that that TP can manipulate on behalf of that user, and must ensure only those executions are performed.
C3. The list of relations in E2 must be certified to meet the separation of duty requirement.
E3. The sysem must authenticate the identity of each user attempting to execute a TP.
C4. All TPs must be certified to write to an append-only CDI (the log) all information necessary to resonstruct the operation.
C5. Any TP taking a UDI as an input must be certified to perform only valid transformations, else no transformations, for any possible value of the UDI. The transformation should take the input from a UDI to a CDI, or the UDI is rejected (typically, for edits as the keyboard is a UDI).
E4. Only the agent permitted to certify entities may change the list of such entities associated with a TP. An agent that can certify an entity may not have any execute rights with respect to that entity.

Notes for November 18, 1998

1. Greetings and Felicitations!
2. Puzzle of the Day
3. 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
4. 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*: $R = E_1(E_2(...E_n(K)...))$
5. Mandatory vs. Discretionary;
   a. security levels
   b. categories
6. 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.
7. Lattice Model
   a. Set of classes *SC* is a partially ordered set under relation ≤ with GLB (⊗), LUB (⊕)
   b. Note: ≤ is reflexive, transitive, antisymmetric
   c. Application to MLS: forms a lattice with elements being the Cartesian product of the linear lattice of levels and the subset lattics of categories
   d. Examples: (A, C) ≤ (A', C') iff A ≤ A' and C ⊆ C';
      (A, C) ⊕ (A', C") = (max(A, A'), C ∪ C')
      (A, C) ⊗ (A', C') = (min(A, A'), C ∩ C')
8. Biba: mathematical dual of BLP
   a. P may read O if L(P) ≤ L(O) and C(P) ⊆ C(O)
   b. P may write O if L(O) ≤ L(P) and C(O) ⊆ C(P)
   c. Combined with BLP: continue example
9. Clark-Wilson
   a. Theme: military model does not provide enough controls for commercial fraud, *etc*. because it does not cover the right aspects of integrity
   b. Data items: "Constrained Data Items" (CDI) to which the model applies, "Unconstrained Data Items (UDIs) to which no integrity checks are applied, "Integrity Verification Procedures" (IVP) that verify conformance to the integrity spec when IVP is run, "Transaction Procedures" (TP) takes system from one well-formed state to another
   c. Certification and enforcement rules:
      C1. All IVPs must ensure that all CDIs are in a valid state when the IVP is run
      C2. All TPs must be certified to be valid, and each TP is assocated with a set of CDIs it is authorized to