Outline for February 2, 1999

1. Greetings and felicitations!
   a. Homework #3 will be on web page tomorrow
   b. Please say when make-up classes would be good or bad
2. Secure vs. Precise
   a. Confidentiality only
   b. Assume: output of a function encodes all available information about inputs (such as resource usage, *etc*.)
   c. Protection mechanism: given function *p*, it's a function *m* such that either *m* = *p* for a given set of inputs, or *m* produces an error message
   d. Confidentiality policy: function which checks that the particular inputs are in the authorized set of inputs
   e. Security:*m* is secure iff there is an *m*' such that, for all inputs, *m* = *m'*(*c*(...)), ie, *m*'s values consistent with stated confidentiality policy
   f. Precision: m1, m2 distinct protection mechanisms. m1 as precise as m2 if, for all inputs, m1 = p implies m2 = p. m1 is more precise id there is an input such that m1 = p and m2 ≠ p on that input.
   g. Union: m1 u m2 = m3, where m3 = p iff m1 = p and m2 = p; otherwise, m3 = m1.
3. Non-deducibility and non-interference policies: can these be composed?
   a. Non-deducibility security: prevent one set of users from deducing information from another set of user's actions
   b. Non-interference security: prevent one set of users from being interfered with as a result of another set of user's actions
4. Machine model
   a. Set of users with security labels High, Low
   b. Set of possible sequences of inputs from users and outputs to users
   c. Trace: interleaving of input and output in a "meaningful" order (eg, temporal)
5. Non-deducibility secure
   a. Means: for every security label *x* and defined trace, there is a second trace that exhibits the same behavior that is visible to users with security labels ≤ *x*, but which has no inputs that are not ≤ *x*.
   b. Implies: higher inputs don't affect output from the point of view of lower subject
   c. If system is NDS, low users can't obtain new information if additional high users are allowed to provide inputs
   d. If system is NDS, and low users can determine specific sets of information based on their interpretation of observed behavior, then removal of the high users should not change the information obtainable by low users
   e. Example: high user inputs into system, low user given parity of number of inputs (both high and low). Assumption: low user cannot examine inputs or the information they contain (as must look like high user not even present). But can observe parity of number of inputs (say, by looking at final value of toggle). This gives a covert channel (high can send messages to low 1 bit at a time) and so is not NDS secure. Fix: introduce high-level outputs, add them into parity.
6. Question: if two NDS systems are composed, is the result NDS?
   a. System A: 2 input, output channels; rules follow:
      HIGH USER INPUT: user gives both high, low inputs to left input channel of A; no-one else can cause such inputs;
      SYSTEM-GENERATED OUTPUT: Random high and low outputs to external environment on right channel only; user can't affect it
      ENVIRONMENT INPUT: Arbitrary inputs, high and low, on right; unaffected by other inputs, and no effect on output
      LOW USER OUTPUT: Low level user can observe low-level inputs, outputs of A; can never see high inputs, outputs
      STOP TERMINATION: System A reaches klow level output called STOP. Only 1 more piece of I/O.
      PARITY OUTPUT: On STOP, low output is number of high-level inputs and outputs.
   b. System A examples:

- One low output (STOP), final low output EVEN (0 high inputs, outputs)
- 1 low input, 1 high input, 2 low outputs (inc. STOP), final low output ODD
- 2 high inputs, 1 low input, 1 low output (STOP), low final output EVEN

c. System B: line System A, but all input into left; none on right, and otherwise right and left are exchanged (eg., B gets STOP input on left).
d. Note: A, B are NDS as a high-level user can't pass info to low-level user; uncertainty of inuts from environment make this so
e. Compose them. A on left, B on ight, left/right inputs/outputs hooked together in the obvious way. Assumptions:
   - A sends STOP to B; atomic in transit, ie, no subsequent output of B sent to A after STOP sent but before it is received
   - right channels of A hooked to left channels of B, ie, no stray inputs, outputs
f. Problem: high user passes high inputs to left channel of A; low level user observes parity outputs of A and B. By symmetry, parity between A and B is ALWAYS 0 (as each input has a corresponding output incident on A or B, and vice versa). Hence if observed parity valuues for A and B are the same, even number of inputs to left channel of A; if parity values differ, the number of inputs is odd. Hence not NDS.

7. Non-Interference Security System Model
   a. $\{ u_1, ..., u_n \}$ users with security labels (HIGH or LOW)
   b. $\{ w_1, ..., w_n \}$ inputs from user i
   c. W is single stream input merged from all n inputs in $\{ w_1, ..., w_n \}$
   d. $\{ [w, 1], ..., [w, n] \}$ of n output sequences , where $[w, i]$ is the output seen by user i with respect to the merged input stream w
   e. $PG_j([w, i])$ is output sequence resulting from input sequence w with all input from u_j removed
   f. System is NIS if for all users, the output sequence is the same as the output sequence purged of input from higher sources, ie.,
      $\text{label}(u_i) > \text{label}(u_j) \Rightarrow [w, j] = PG_i([w, j])$
8. Example
   a. Same system as for NDS.
   b. High-level user does nothing; output is always 1 (as input parity is even; none)
   c. High-level user does something; output may be 0 as input may be even or odd.
   d. Moral: not NIS. Can close off output, or add random inputs, to end this.
9. Question: if two NIS systems are composed, is the result NIS?
   a. High user gives messages to system which must respond with output messages (*i.e.*, ack them); may also receive high messages from another source, must ack them too
   b. After some time, issues output message of parity of high-level messages, and ignores all following
   c. Compose 2 such systems
   d. If high user does nothing, no acks sent and low user will see all messages acknowledged. If high user sends message, systems send msgs between themselves, and at shutdown at least 1 message not acknowledged. Different outputs.