Notes for November 8, 1999

1. Greetings and Felicitations!
2. Puzzle of the Day
3. Password Storage
   a. In the clear; MULTICS story
   b. Enciphers; key must be kept available; get to it and it's all over
   c. Hashed; present idea of one-way functions using identity and sum
   d. Show UNIX version
4. Attack Schemes Directed to the Passwords
   a. Exhaustive search: UNIX is 1-8 chars, say 96 possibles; it's about 7e16
   b. Inspired guessing: think of what people would like (see above)
   c. Random guessing: can't defend against it; bad login messages aid it
   d. Scavenging: passwords often typed where they might be recorded (b\as login name, in other contexts, *etc*.
   e. Ask the user: very common with some public access services
   f. Expected time to guess
5. Password aging
   a. Pick age so when password is guessed, it's no longer valid
   b. Implementation: track previous passwords vs. upper, lower time bounds
6. Ultimate in aging: One-Time Pads
   a. Password is valid for only one use
   b. May work from list, or new password may be generated from old by a function
   c. Example: S/Key™
7. Challenge-response systems
   a. Computer issues challenge, user presents response to verify secret information known/item possessed
   b. Example operations: $f(x) = x+1$, random, string (for users without computers), time of day, computer sends $E(x)$, you answer $E(D(E(x))+1)$
   c. Note: password never sent on wire or network
   d. Attack: monkey-in-the-middle
   e. Defense: mutual authentication (will discuss more sophisticated network-based protocols later)
8. Biometrics
   a. Depend on physical characteristics
   b. Examples: pattern of typing (remarkably effective), retinal scans, *etc*.
9. Location
   a. Bind user to some location detection device (human, GPS)
   b. Authenticate by location of the device