Notes for January 23, 1998

1. Greetings and felicitations!
   a. **Reading**: Pfleeger, pp. 91–118
2. Puzzle
   a. Point is that PGP relies on 2 keys, derives strength from weakest of them
3. RSA
   a. Provides both authenticity and confidentiality
   b. Go through algorithm:
   Idea: $C = M^e \bmod n$, $M = C^d \bmod n$, with $ed \bmod \phi(n) = 1$.
   Proof: $M^{\phi(n)} \bmod n = 1$ [by Fermat's theorem as generalized by Euler]; follows immediately from $ed \bmod \phi(n) = 1$.
   Public key is (e, n); private key is d. Choose $n = pq$; then $\phi(n) = (p–1)(q–1)$.
   c. Example:
   $p = 5$, $q = 7$; $n = 35$, $f(n) = (5–1)(7–1) = 24$. Pick $d = 11$. Then $de \bmod \phi(n) = 1$, so choose $e = 11$. To encipher 2, $C = M^e \bmod n = 2^{11} \bmod 35 = 2048 \bmod 35 = 18$, and $M = C^d \bmod n = 18^{11} \bmod 35 = 2$.
   d. Example: $p = 53$, $q = 61$, $n = 3233$, $f(n) = (53–1)(61–1) = 3120$. Take $d = 791$; then $e = 71$.
   Encipher M = RENAISSANCE: A = 00, B = 01, …, Z = 25, blank = 26. Then:
   M = RE NA IS SA NC Eblank = 1704 1300 0818 1800 1302 0426
   $C = (1704)^{71} \bmod 3233 = 3106$; *etc.* = 3106 0100 0931 2691 1984 2927

[*ended here*]

4. Authentication:
   a. validating client (user) identity
   b. validating server (system) identity
   c. validating both (mutual authentication)
5. Basis
   a. What you know
   b. What you have
   c. What you are
6. Passwords
   a. How UNIX does selection
   b. Problem: common passwords; Go through Morris and Thompson ; Klein and mine, *etc.*
   c. May be pass phrases: goal is to make search space as large as possible and distribution as uniform as possible
   d. Other ways to force good password selection: random, pronounceable,m computer-aided selection

   e. Go through problems, approaches to each, *esp*. proactive
7. Password Storage
   a. In the clear; MULTICS story
   b. Encipheres; 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
8. 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
9. 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
10. 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
11. 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)
12. Biometrics
   a. Depend on physical characteristics
   b. Examples: pattern of typing (remarkably effective), retinal scans, etc.
13. Location
   a. Bind user to some location detection device (human, GPS)
   b. Authenticate by location of the device