Outline for January 14, 1999

1. Greetings and felicitations!
   a. *lassen* not reachable from 169.237 network; we're trying to figure it out. You can get to it from any 128.120 host, such as *toadflax*
2. Quote for the Day
3. Simple substitution ciphers
   a. Go through rotor systems (Enigma)
   b. Go through one-time pads
4. Polygram substitution cipher: Playfair
   a. Key is 5 by 5 grid
   b. If m_1, m_2 in same row, c_1, c_2 are to immediate right of m_1, m_2 respectively
   c. If m_1, m_2 in same column, c_1, c_2 are below m_1, m_2
   d. If $m_1 = m_2$, insert null in text
   e. if m_1, m_2 in different rows and columns, c_1, c_2 form rectangle with c_1 in m_1's row and c_2 in m_2's row
   f. if odd number of characters in message, append null.
5. History
   a. IBM did Lucifer, submitted it in response to NIST CFP
   b. NIST (really, NSA) suggested some minor changes; major one was to make key 56 bits, not 112.
6. Show the cipher
   a. Product cipher with 64 bits in, 64 bits out, and 16 48-bit round keys generated from 56 bit key
   b. Note S-boxes are real heart of algorithm
7. Known attacks and weaknesses
   a. Complementation property: $DES_k(m) = (DES_k(m'))'$ where x' is the bitwise complement of x;
   b. Weak, semiweak keys
   c. If it's a group, multiple encipherment worthless (as group is closed under composition)
   d. Differential cryptanalysis: first version unusable as at 16 rounds, more plaintext/ciphertext pairs needed than exhaustive key trial; but for 15 rounds, cuts this time. Later versions cut it to 2^{47} tries. Works by comparing xors of results with xors of corresponding plaintext.. Designers of DES knew about this one, hence the design of the S-boxes
   e. Linear cryptanalysis drops required chosen plaintext/ciphertext pairs to 2^{42}; not known to designers of DES.
8. DES Modes
   a. ECB
   b. CBC
   c. note that OFB and CFB exist, essentially use DES as a pseudorandom bitstream generator; OFB feeds back before xor, CFB after
   d. Triple DES and EDE mode
9. Public Key Ciphers
   a. 2 keys, public k and private p; public key published, private key kept secret
   b. Requires: (1) given k, computationally infeasible to compute p; (2) immune to chosen plaintext attack (as adversary can generate any amount of ciphertext); (3) given C and k, computationally infeasible to find M
   c. Show confidentiality, authenticity; emphasize integrity is by-product of authenticity, except that one needs to keep the order of blocks correct
   d. Some ciphers do confidentiality and authenticity, others do only one
10. Do RSA
    a. Exponentiation cipher: $C = m^e \bmod n$, $M = C^d \bmod n$; d is private key, (e, n) is public key; must choose d first, then e so that $ed \bmod \phi(n) = 1$.
    b. Why? as $ed \bmod \phi(n) = 1$, $ed = t\phi(n) + 1$ for some integer t. Then
    $C^d \bmod n = (M^e \bmod n)^d \bmod n$
    $= M^{ed} \bmod n$

$= M^{t\phi(n)+1} \bmod n$
$= M(M^{t\phi(n)} \bmod n) \bmod n$
$= \mathrm{M}(M^{\phi(n)} \bmod n)^t \bmod n$
$= M(1)^t \bmod n$
$= M \bmod n$

by Fermat's Little Theorem

c. Example: $p = 5$, $q = 7$, $n = 35$, $\phi(n) = 24$; choose $e = 11$, then $d = 11$. HELLO WORLD is 07 04 11 11 14 22 14 17 11 03; enciphering is $C = 07^{11} \bmod 35 = 28$, *etc*. so encipherment is 28 09 16 16 14 08 14 33 16 12.

11. ACM and primitive operations
    a. Go over subjects, objects (includes subjects), and state (S, O, A) where A is ACM
    b. Transitions modify ACM entries; primitive operations follow
    c. **enter** *r* **into** $A[s,o]$
    d. **delete** *r* **from** $A[s,o]$
    e. **create subject** *s'* (note $A[s',x] = A[x,s'] = \emptyset$ for all x)
    f. **create object** *o'* (note $A[x,o'] = \emptyset$ for all x)
    g. **destroy subject** *s'*
    h. **destroy object** *o'*
12. commands
    a. **command** $c(s_1, ..., s_k, o_1, ..., o_k)$
    if r_1 **in** $A[s_1, o_1]$ **and**
    r_2 **in** $A[s_2, o_2]$ **and**
    ...
    r_m **in** $A[s_m, o_m]$
    then
    op_1;
    op_2;
    ...;
    op_n;
    end.
    b. Example 1: creating a file
    command *create_file*(p, f)
    create object *f*;
    enter *Own* **into** $A[p, f]$
    enter *Read* **into** $A[p, f]$
    enter *Write* **into** $A[p, f]$
    end.
    c. Example 2:granting one process read rights to a file
    command *grant_read*(p, q, f)
    if *Own* **in** $A[p, f]$
    then
    enter *Read* **into** $A[q, f]$
    end.
13. What is the safety question?
    a. An unauthorized state is one in which a generic right *r* could be leaked into an entry in the ACM that did not previously contain *r*. An initial state is safe for *r* if it cannot lead to a state in which *r* could be leaked.
    b. Question: in a given arbitrary protection system, is safety decidable?
14. Mono-operational protection systems: decidable
    a. Theorem: there is an algorithm that decides whether a given mono-operational system and initial state is safe for a given generic right.
    b. Proof: finite number of command sequences; can eliminate **delete** , **destroy** .
    Ignore more than one **create** as all others are conditioned on access rights in the matrix. (One exception: no subjects; then we need one **create subject**).
    Bound: *s* number of subjects (possibly one more than in original), *o* number of objects (same), *g* number of

generic rights; number of command sequences to inspect is at most 2^{gso}.

Quote

"Without harmony in the state, no military expedition can be undertaken; without harmony in the army, no battle array can be formed."

— Sun Tzu, *The Art of War*, (Translated by James Clavell), Dell Publishing, New York, NY 10036 (1983).