SECURITY

SECURITY ISSUES:

External protection of a system. A classified site goes to extraordinary lengths to keep things physically tight. Among the issues to be considered:

- **Unauthorized access** Mechanism assuring only authorized individuals see classified materials.
- **Malicious** modification or destruction
- **Accidental** introduction of inconsistency.
- **Authentication** How do we know the user is who she says she is. Can have passwords on domains.
- **Protection of passwords** is difficult. Issues include:

  a) It's very easy to guess passwords since people use simple and easily remembered words.

  b) Need exists to change passwords continually.

  c) Limiting number of tries before locking up.

  d) How to crack UNIX passwords.

Trojan Horse: A piece of code that misuses its environment. The program seems innocent enough, however when executed, unexpected behavior occurs.

Trap Doors: Inserting a method of breaching security in a system. For instance, some secret set of inputs to a program might provide special privileges.

Threat monitoring:	Look for unusual activity. Once access is gained, how do you identify someone acting in an unusual fashion?
Audit Log:	Record time, user, and type of access on all objects. Trace problems back to source.
Worms	Use spawning mechanism; standalone programs.
Internet Worm:	In the Internet worm, Robert Morse exploited UNIX networking features (remote access) as well as bugs in finger and sendmail programs. Grappling hook program uploaded main worm program.
Viruses	Fragment of code embedded in a legitimate program. Mainly effects microcomputer systems. These are often downloaded from public bulletin boards, or via the exchange of floppies.
Firewall	A mechanism that allows only certain traffic between trusted and untrusted systems. Often applied to a way to keep unwanted internet traffic away from a system.

CRYPTOGRAPHY

ATTACK METHODS:

- Attacks on a distributed system include:

  Passive wiretapping. (unauthorized interception/reading of messages)
  Active wiretapping:

  a) **Modification** - changing a portion of the message.

  b) **Spurious messages** - introducing bogus messages with valid addresses and consistency criteria.

  c) **Site impersonation** - claiming to be some other logical node.

  d) **Replay** of previous transmission - repeating previous valid messages. (for example, authorization of cash withdrawal.)

- Cryptography is the only known way to prevent these attacks.

DEFINITIONS:

Encryption:

C = E(M, Ke)

E = Encyphering Algorithm

M = Message - plain text

Ke = Encryption key

C = Cyphered text

Decryption:

M = D(C, Kd)

D = Decyphering Algorithm

Kd = Decryption key

Cryptosystems are Conventional or Public Key

- Conventional is symmetric; Ke = Kd , so the key must be kept secret. Algorithms are simple to describe, but complex in the number of operations.

- Public key is asymmetric; Ke != Kd , so Ke can be made public. Kd is secret and can't easily be derived from Ke .

Security against attack is either:

- **Unconditionally secure** - Ke can't be determined regardless of available computational power.

- **Computationally secure:** - calculation of Kd is economically unfeasible (it would overwhelm all available computing facilities.)

- The only known unconditionally secure system in common use involves a random key that has the same length as the plaintext to be encrypted.
- The key is used once and then discarded. The key is exclusively OR'd with the message to produce the cypher.
- Given the key and the cypher, the receiver uses the same method to reproduce the message.

CONVENTIONAL CRYPTOSYSTEMS

Transposition:

CRY PTO GRA PHY -->
RYC TOP RAG HYP

- This type of code is relatively easy to break given sufficient text; the relative frequency of letters remains the same.

Substitution:

CRY PTO GRA PHY -->
FUB SWR JUD SKB

- This is also easy to break; it becomes more complicated with 1 - to - many mappings and poly-character substitutions.

Feedback:

$$C1 = E(B1, K)$$
$$Ci = E(Bi, K) \text{ XOR } B(i - 1)$$

Each block of cypher is a function of previously encrypted blocks:

DATA ENCRYPTION STANDARD (DES):

- The official National Institute of Standards and Technology (NIST), (formerly the National Bureau of Standards) encryption for use by Federal agencies.

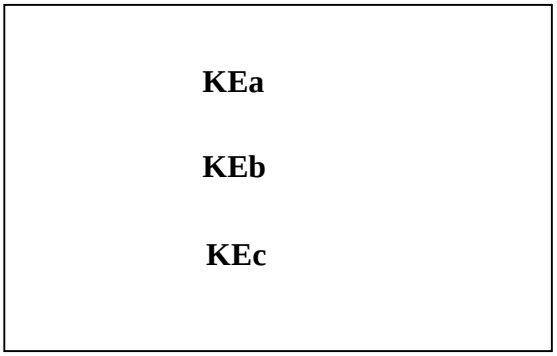
- The source of security is the non-linear many-to-one function applied to a block of data. This function uses transposition and substitution. The algorithm is public, but the key (56 bits) is secret.

- Just how secure is DES? The Feds aren't telling.

- There's concern that a cryptoanalyist might be able to do a brute force calculation of the 56 bit key. The counter-argument is that the method is sound, and the key can simply be made longer.

PUBLIC KEY CRYPTOSYSTEMS:

The general principle is this: Any **RECEIVER A** uses an algorithm to calculate an encryption key **KEa** and a decryption key **KDa**. Then the receiver PUBLICIZES **KEa** to anyone who cares to hear. But the receiver keeps secret the decryption key **KDa**. **User B** sends a message to **A** by first encrypting that message using the publicized key for that receiver **A**, **KEa**. Since only **A** knows how to decrypt the message, it's secure.



Public

To be effective, a system must satisfy the following rules:

1. Given plaintext and ciphertext, the problem of determining the keys is computationally complex.

2. It is easy to generate matched pairs of keys Ke, Kd that satisfy the property

$$D(E(M, Ke), Kd) = M.$$

- This implies some sort of trapdoor, such that Ke and Kd can be calculated from first principles, but one can't be derived from the other.

3. The encryption and decryption functions E and D are efficient and easy to use.

4. Given Ke , the problem of determining Ke is computationally complex.
   - What is computationally difficult? Problems that can't easily be calculated in a finite time.
   - Examples include: factoring the product of two very large prime numbers; the knapsack problem.
   - These problems are NP complete - solution times are exponential in the size of the sample.

5. For almost all messages it must be computationally unfeasible to find ciphertext key pairs that will produce the message.
   - (In other words, an attacker is forced to discover the true (M,Ke) pair that was used to create the ciphertext C.)

6. Decryption is the inverse of encryption.

$$E(D(M, Kd), Ke) = D(E(M, Ke), Kd)$$

AN EXAMPLE:

1. Two large prime numbers p and q are selected using some efficient test for primality. These numbers are secret:

   Let p = 3, q = 11

2. The product n = p * q is computed.

   n = 3 * 11 = 33.

3. The number Kd > max(p, q) is picked at random from the set of integers that are relatively prime to and less than L(n) = (p - 1) (q - 1).

   L(n) = (p - 1) (q - 1) = 20.
   Choose Kd > 11 and prime to 20.
   Choose Kd = 13.

4. The integer Ke , 0 < Ke < L(n) is computed from L(n) and Kd such that Ke * Kd = 1 (mod L(n)).

   0 < Ke < 20
   Ke = 17. (since 17 * 13 = 221 = 1 (mod 20))

5. Separate the text to be encoded into chunks with values 0 - (n - 1).

   In our example, we'll use < space = 0, A = 1, B = 2, C = 3, D = 4, E = 5 >.

   Then " B A D <sp> B E E " --> "21 04 00 25 05"

21 ^ 17 (mod 33)	= 21.	21 ^ 13 (mod 33)	= 21.
04 ^ 17 (mod 33)	= 16.	16 ^ 13 (mod 33)	= 04.
00 ^ 17 (mod 33)	= 00.	00 ^ 13 (mod 33)	= 00.
25 ^ 17 (mod 33)	= 31.	31 ^ 13 (mod 33)	= 25.
05 ^ 17 (mod 33)	= 14.	14 ^ 13 (mod 33)	= 05.

This whole operation works because, though n and Ke are known, p and q are not public. Thus Kd is hard to guess.

[Note: recently a 100 digit number was successfully factored into two prime numbers.]

AUTHENTICATION AND DIGITAL SIGNATURES:

Sender Authentication:

In a public key system, how does the receiver know who sent a message (since the receiver's encryption key is public)?

Suppose **A** sends message **M** to **B**:

a) **A** DECRYPTS **M** using **A's Kd(A)** .

b) **A** attaches its identification to the message.

c) **A** ENCRYPTS the entire message using **B's** encryption, **Ke(B)**

C = E ((A, D(M, Kd(A))), Ke(B))

d) B decrypts using its private key **Kd(A)** to produce the pair **A, D(M, Kd(A)).**

e) Since the proclaimed sender is **A**, **B** knows to use the public encryption key **Ke(A)**.

Capture/Replay

- In this case, a third party could capture / replay a message.
- The solution is to use a rapidly changing value such as time or a sequence number as part of the message.