ENCRYPTION

Today

- The last class described a number of problems in ensuring your security and privacy when using a computer on-line.
- This lecture discusses one of the main technological solutions.
- The use of cryptography.
- How a book could be classified as an armament.

Basic principles of cryptography

- Messages are put into code (encrypted) by the sender and decoded (decrypted) by the receiver.
- Basic ingredients of conventional cryptography:
  - Plain text input
  - Encryption algorithm
  - Secret key shared by sender and recipient
  - Cipher text (coded input text)
  - Decryption algorithm

Cryptography example

- Suppose input text is
  THE SKY IS BLUE
- Algorithm:
  Replace each letter by the letter in the alphabet 1 step along.
- Output:
  UIF TLZ JT CMVF
- The process used here is called *substitution* — substituting one element (in this case a letter) by another.
- Another process is *transposition* — Moving parts of the message around, e.g.
  TLZ UIF JT CMVF

Requirements

- An encryption algorithm and a decryption algorithm are required.
- Ideally, we would like a strong encryption algorithm, secure against attack.

  An opponent should be unable to decrypt the ciphertext or discover the key even if s/he is in possession of a number of ciphertexts together with the plain text which produced them.
- Both sender and receiver must have the secret key(s) for the process to work.
- Note: the security of conventional encryption depends on the secrecy of the key, not secrecy of the algorithm.

Classification of cryptographic systems

- The type of operations used to transform plaintext to ciphertext:
  - *substitution*
  - *transposition*
  - Usually some complex combination of these is used.
  - In any case, no information can be lost in the process.
- Whether sender and receiver use the same keys
  - *symmetric*: sender and receiver use the same keys
  - *asymmetric*: sender and receiver use different keys

Classification of cryptographic systems (2)

- The number of keys used
- How the plaintext is processed.
  - A *block cipher* processes the input one block of elements at a time, producing an output block for each input block.
  - A *stream cipher* processes the input elements continuously, producing one element at a time as it goes along.

Cryptanalysis

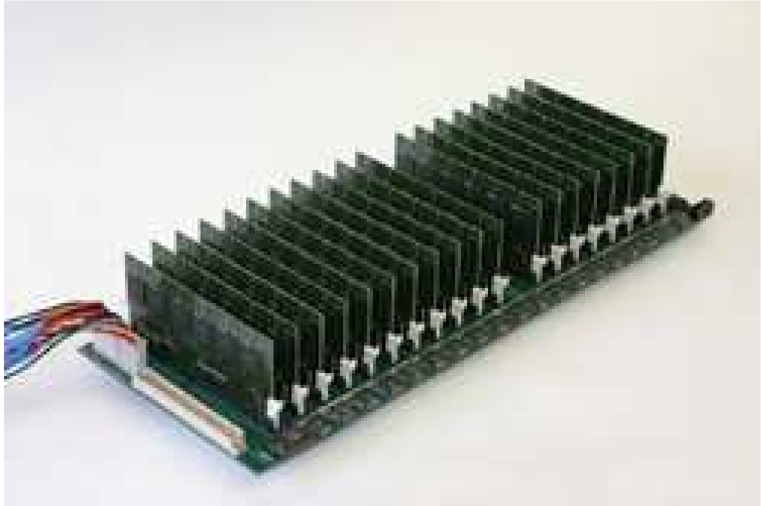
- The process of attempting to discover the plaintext or the key.
- Known plain text attack:
  - The opponent has a sample of plaintext and ciphertext, and from this infers the keys; e.g., he may use brute force to try lots of different keys until successful.
  - Note that plain text may be compressed and may be numerical in origin, so brute force methods usually require some knowledge of the type of plain text used.
  - For a key of length 128 bits, it would take an opponent about 10^{18} years to crack!

Cryptanalysis (2)

- chosen plain text attack:
  - The opponent gets the computer doing the encryption to encrypt some specially-chosen text.
- differential cryptanalysis attack:
  - The opponent gets the computer doing the encryption to encrypt several blocks of text which differ only slightly.
- differential fault analysis:
  - The opponent attacks the hardware of the encryption computer to force it to make mistakes, in order to discover the key or algorithm.

Some symmetric algorithms

- Data Encryption Standard (DES)
  - US Government standard in use 1977–1998.
  - Algorithm works on blocks of data (each 64-bits), and uses a 56-bit key.
  - Decertified in 1998 and replaced by 3DEA.
  - Possible to crack with brute force.



Some symmetric algorithms (2)

- Triple Data Encryption Algorithm (3DEA)
  - Proposed in 1979, became a standard in 1999.
  - Applies the DES algorithm 3 times to plain text:
    Encrypt with Key A, decrypt with Key B, encrypt with Key C.
    Decryption is the reverse with the keys reversed: Decrypt with Key C, encrypt with Key B, decrypt with Key A.
  - Each key of length 56 bits, so effective key length of 168 bits.

Weaknesses?

Asymmetric (or public) key algorithms

- Similar to symmetric key encryption, but we use at least 2 keys: One for encryption (the public key), and one for decryption (the private key).
- The steps involved are:
  - Keys are generated in pairs, a public key and a private key, by each person or computer (say Bob).
  - The public key is made public (e.g., on a web-site) by Bob.
  - a message to Bob uses his public key.
  - Bob decodes the message using his private key.
- This approach dates from 1976.

Trapdoor functions

Public key algorithms

- RSA algorithm
  - Developed by Rivest, Shamir & Adleman at MIT in 1977
  - RSA is a block cipher in which the plaintext and cipher text are integers between 0 and $(n-1)$ for some n.

    $n = pq$ where p and q are large prime numbers

    If M is the plaintext number, and C is the cipher text number, the algorithm works as follows:

    Encryption algorithm: $C = M^e \; modulo \; n$

    Decryption algorithm: $M = C^d \; modulo \; n$

Public key algorithms (2)

- RSA algorithm (cont)
  - Both sender and receiver must know the values of n and e. The public key is a pair of numbers (e, n).
  - Only the receiver knows the value of d. The private key is the pair of numbers (d, n).
  - Secure because d is determined from p and q, and it is hard to compute these from n.
- Digital Signature Standard (DSS)
  - A standard agreed in 1993 for digital signatures in the American National Institute of Standards.
  - Only used for digital signatures (not for encryption or key exchange).

Requirements for public key algorithms

- It is computationally easy for party B to generate a pair of keys.
- It is computationally easy for sender A to generate the cipher text on the basis of the plain text and the public key.
- It is computationally easy for party B to decrypt the resulting ciphertext using his private key and so generate the plain text.
- It is computationally infeasible for an opponent to determine the private key from the public key.
- It is computationally infeasible for an opponent to recover the original plain text from the public key and the ciphertext.
- In addition, we may require (not necessary but nice to have): Either of the two related keys may be used for encryption with the other used for decryption.

Applications of public key methods

- Encryption —
  sending coded messages.
- Authentication —
  when we want to be certain that the sender of a message is actually the person (or computer) they say they are. The sender of the message uses his private key to encrypt the message. Only his public key will be able to decode the message.
- Digital Signature —
  The sender “signs” a message using his private key. This application is similar to authentication.
- Key Exchange —
  Two parties co-operate to exchange a session key, using the private key of one or both parties.

How to distribute public keys?

How to distribute public keys?

- Answer is simple: put on your web-site, email your friends, shout it from the roof-tops!
- But if Alice gets an email from Bob telling her that 1023 is his public key, how does she know it really is his? Maybe someone is impersonating him and sending out a false key in his name!
- Digital Certificates seek to get around this. A user (e.g., Bob) presents his public key to a trusted third party and receives a digital certificate. The certificate contains a public key together with a a user ID for the key owner (Bob), all signed by the third party.
- Examples of third parties: Government agencies or a bank. The user (Bob) can then give the digital certificate to anyone else (e.g., Alice).

Public key distribution of symmetric keys

- How do 2 parties share a symmetric (secret) key?
- They could deliver them physically (e.g., by courier).
- If they already share a secret key, they could send the new one by encrypted message.
- They could use public key certificates, as follows:
  1. Bob sends Alice his public key using a public key certificate.
  2. Alice encrypts a message using one-off symmetric key for this session (a *session key*).
  3. Alice encrypts the session key using Bob's public key.
  4. Alice attaches the encrypted session key to the message and sends it to Bob.

Only Bob is able to decrypt the session key (since only he has his private key). So, only Bob can read the original message.

Cryptographic systems (1)

Cryptography is used in several systems serving a variety of purposes

- Message Digest Functions
  - These produce a summary digest of a file, and can be used to see if the file has been altered.
  - Useful for detecting presence of viruses or tampering by opponents.
  - Sometimes used for message authentication codes (appended to a message, so that the receiver can see if the message was altered during transit).
  - Examples: HMAC, MD series (128 bit digest), SHA series (160 bit digest).

Cryptographic systems (2)

- Digital signatures
  - Unique identifier of a sender of a message
  - Can use public key cryptography in reverse
- Digital Certificates
  - Issued by trusted third party (e.g. bank, government agency) to verify user is who they say they are.
  - Usually third party's signature encrypted by the private key of the authorization party
    - So the receiver (Bob) needs to decode the authorization signature using the public key of the third party.
    - Then, if this works, use the public key of the sender (Alice) to decode her signature.

Cryptographic systems (3)

- “Pretty Good Privacy” (PGP)
  - A publicly-available system for encrypting files and email messages
  - PGP uses:
    - RSA for management of keys in symmetric encryption
    - IDEA algorithm for sending data using symmetric encryption
    - MD5 scheme for ensuring no tampering.
  - Main weakness: if a public key is compromised, than a revocation certificate has to be issued to everyone in contact with the person whose keys are compromised.

Summary

- This lecture discussed some basic aspects of the use of encryption.
- We talked about:
  - Private key encryption
  - Public key encryption