Cryptography

cryptography is used to fulfil the following functions

- **confidentiality**: information in a message is only accessible by those people authorised to access it
- **authentication**: it should be possible for the receiver of a message to know who really sent the message
- **integrity**: it should be possible for the receiver of a message to verify that the message has not been modified in transit
- **non-repudiation**: a sender should not be able to falsely deny that they sent the message

---

A Simple Cipher

the Caesar cipher is named after Julius Caesar, said to be the first to use it

in the Caesar cipher each character is substituted by another

a technique called a mono-alphabetic cipher

Thus using a Caesar cipher, the message

- "Mission Impossible" would be encoded as:
- "Awggweb Wadgwpzs"

---

Types of cipher

- **simple substitution cipher**
  - mono-alphabetic, where each character in the plain text is replaced with a corresponding character of cipher-text

- **homo-phonic substitution cipher**
  - is like a simple substitution cipher except that a single character of plain-text can map onto several characters of cipher-text
  - A might map onto 5, 14, 147
  - used in 1401 by Duchy of Mantua

---

Types of ciphers

- **symmetric cipher**
  - same key used to encrypt and decrypt information

- **asymmetric cipher**
  - does not use the same key to encrypt and decrypt information
Types of cipher

- polygram substitution
  - blocks of plain-text characters are encrypted in groups into blocks of cipher-text
  - ABA might map onto RTQ
  - used by British during WWI

- polyalphabetic substitution
  - cipher is made up of multiple mono-alphabetic ciphers
  - the cipher changes with the position of each character in the plain text
  - used in American civil war by the Union

Cryptanalysis and attacks

- known plain text attack
  - analyst has cipher-text and plain-text
  - tries to deduce the keys

- chosen plain text attack
  - has all plain text attack and also can choose the particular plain-text to be encrypted
  - tries to deduce the keys

- adaptive chosen plain text attack
  - special case of chosen plain text attack
  - can modify the choice of plain-text to be encrypted based on previous results
  - tries to deduce the keys

- purchase key attacks
  - analyst threatens, blackmails, tortures or bribes someone for the keys.

- cipher-text only attack
  - analyst has cipher-text of several messages
  - all encrypted using same encryption algorithm and key
Modern encryption algorithms

- DES was created in 1977
  - plain-text divided into 64 bits in length and the key is 56 bits long
  - pros: can be implemented in hardware
  - cons: can be broken in 3.5 hours on a fast machine
    - watch this space for breaking DES on the new 64 bit hardware
  - DES is a monoalphabetical substitution cipher using a 64 bit character
    - fundamentally a 56 bit key is too short and this algorithm is open to a brute force attack

Triple DES

- with triple DES, each block of plain-text is processed three times
- pros: two key version of DES is estimated to be $10^{13} \times$ stronger
  - the triple DES even stronger
    - triple DES uses two keys, K1 used, K2 used, then K1 used again

IDEA

- should be used instead of DES or triple DES
- it uses a 128 bit key
- input is characters of 64 bits
- no currently known technique is thought to break IDEA
  - keysize makes it secure from brute force for many decades
  - Tanenbaum 1997, Computer Networks, p596

Triple DES

- cons: if key1, key2 are the same triple DES collapses to single DES
  - key management is harder

![Triple DES Diagram]
Public/Private key encryption

- Key management is a problem with previous algorithms
  - How does someone send their key to the receiver?
- Algorithms have been discovered which use two keys
  - Private
  - Public

Combining Public/Private key encryption and signatures

- Suppose that Alice wishes to send a message to Bob
- Bob wishes to know that the message really comes from Alice
- The message must only be read by Bob and Alice

Combining Public/Private key encryption and signatures

- Both Alice and Bob publish their public keys
  - Alice = Apub
  - Bob = Bpub
- And keep their private keys hidden
  - Alice = Apri
  - Bob = Bpri

Public/Private key encryption

- Public/private key encryption is asymmetric
- Each user publishes their public key
  - So: $P = D(K_{priv}, E(K_{pub}, P))$
    - A user can decode with a private key what someone else has encoded with the corresponding public key
  - Also $P = D(K_{pub}, E(K_{priv}, P))$
    - A user can decode with a public key what someone else has encrypted with the corresponding private key
    - Used for digital signatures
Combining Public/Private key encryption and signatures

- Alice encrypts a document, M, with her private key
  - $E_{(Apri, M)}$
- she encrypts the document with Bob's public key and sends the message, C, to Bob
  - $C = E_{(Bpub, E_{(Apri, M)})}$

- only Bob can decrypt this message with his private key $i = D_{(Bpri, C)}$
- Bob then decrypts this message with Alice's public key
  - $M = D_{(Apub, i)}$
- unfortunately all this can be done transparently via software!

Some practical scripts: a script to log into remote server

```
#!/usr/bin/expect
log_user 0
spawn pwd
expect -re "^.*$"
set localdir $expect_out(1,string)
expect eof
log_user 0
spawn ssh merlin
expect "Password: "
send "abcdefghi\r"
expect "\"$
send "cd $localdir"
```

```
fred@isode:~/GM2/gcc-3.2/gcc/gm2$ $HOME/SHELL/mcd /home/fred/GM2/gcc-3.2/gcc/gm2
fred@merlin:~/GM2/gcc-3.2/gcc/gm2$
```

Remote enabling of a modem

```
#!/usr/bin/expect
log_user 0
spawn pwd
expect -re "^.*$"
set localdir $expect_out(1,string)
expect eof
log_user 0
spawn ssh -l root merlin
expect "Password: "
send "abcdefghi\r"
expect "\"$
send "cd $localdir"
```

what permissions should these scripts have?

what security risks do they present?