

Security

Peter Levinsky, Roskilde, Datamatiker

28.04.2025

Cryptography

- Why Cryptography
- Symmetric Encryption
 - Key exchange
- Public-Key Cryptography
 - Key exchange
 - Certification

Why Cryptography

- General Security Goal
 - Confidentiality
 - Message Integrity
 - End-point Authentication
 - Operational security

Potentially Threats

- Eaves dropping
- Modification, insertions, deletion
- Masquerade
- Playback
- Man-in-a-middle-attack
- DDoS (ie. SYN-flooding)

Cryptography - general

- To send messages over a network,
which is unable to understand for a third part
- General technique:
 - Plain text
 - Encode (algorithm + key)
 - Cipher text (send over the network)
 - Decode (algorithm + key)
 - Plain text

Cryptography - coding

- Old days simple letter transformation
e.g. **c** for an **a** and **d** for a **b** and so on
i.e. **ape** -> **crg**
- To day
 - Symmetric key
 - Public-private keys

How to break encryption

- Force brute
 - Try combinations
 - Ciphertext-only attack – try all
 - Known-plaintext Attack – know few word in text
 - Chosen-plaintext Attack – know one full text

Cryptography - Symmetric

- Using the same key to **encode** and **decode**
- Which goal are fulfilled ?
 - Confidentiality
 - Authentication
 - Integrity
 - Operational security

-Confidentiality – yes
- Authentication - yes
- Integrity - yes
- Operational security - no

Cryptography - Symmetric

- Implementations:
 - DES (Data Encryption Standard) most known
(round 4 min)
today modified to triple DES
key length 64bit (3*64 bit)
 - Other IDEA, RC5
to day even 128 bit (round 149 trillion year)

Cryptography - Symmetric

- Key exchange:
 - Problem to exchange the key
 - we can not send it with a mail
 - then it would not be secret any longer
 - Using a Key Distribution Center (KDC)
 - I have an agreement with the KDC and with this an secret key.
 - So have all I communicate with.

Cryptography - Symmetric

- Windows use Symmetric key in an implementation called Kerberos:
 - All like KDC but you get grant (a key) to a resource for a certain time (all called a ticket)

Cryptography Asymmetric

- Using the different keys to **encode** and **decode**
- You always have a pair of keys
a public key and a private key
- If you **encode** with a **public key** – you must **decode** with a **private key**
- If you **encode** with a **private key** – you must **decode** with a **public key**

Cryptography Asymmetric

Which goal are fulfilled from A to B (B public Key)?

- Confidentiality
- Authentication
- Integrity
- Operational security

-Confidentiality – yes
-Authentication - no
- Integrity - no
- Operational security - no

Cryptography Asymmetric

Which goal are fulfilled from A to B (A private Key)?

- Confidentiality
- Authentication
- Integrity
- Operational security

- Confidentiality – no
- Authentication - yes
- Integrity - yes
- Operational Security - no

Cryptography Asymmetric

Can we fulfilled both

- Confidentiality and
 - Authentication
 - Integrity
- ?

YES – **encode** with **A private key** and then with **B public key** i.e. twice

Cryptography Asymmetric

- Implementations:
 - RSA – most known
key length recommended 1024bit (2048bit)
 - (512 bit brute force approx. 5 month)

Cryptography Asymmetric

- Key exchange:
 - Problem to exchange the key
 - public key are public to everyone
 - ***But do we believe the sender of the key***
 - Using Certification
 - I believe in some Certification Authorities
e.g. VeriSign, Thrust, (in DK TDC)
 - get the public key from one of those trusted third part companies.

Cryptography Asymmetric

- To fulfilled the goal you must encode 2 times (A private and B public)
- A more easy way is to create a Message Digest (MD) a sort of a checksum
- And this 'checksum' are encoded with A's private key (Digital Signature). Then the whole message + the MD are encoded with B's public key

Cryptography Asymmetric

- For authentication :
 - Message Authentication Code (MAC)
 - Both sides – shared secret (s)
 - Send $m + H(m+s)$
 - Check $m+s$ hashed $== H(m+s)$
 - Fill with Nonsens

Cryptography Mixed

- Using asymmetric keys to exchange a symmetric key
- Then use this symmetric key for rest of this session.

This increase the speed of encryption and decryption.