Cryptography; class 3

30/01/2019

Baptiste HAMON

# Definitions

Storing and transmitting data

Kerckhoff’s Principle : Only secrecy involved in the system should be the key, the algorithm should be publicly known. If more can see the code and test it, the more weaknesses will be spotted

Types of cipher :

* Substitution : replaces bits, characters, blocks of characters with different bits.
* Transposition : does not replace the text but move it around, keep the integrity of the data

**Confusion** : Need a complex algorithm to go from plaintext to ciphertext → there must be no simple relationship between them.

**Diffusion** : single plaintext bit has influence over several of the ciphertext bits. Used a lot in the hashing algorithm.

While using an hashing algo :

* md5(text) ⇒ single cipher text with a specific length = hash. You can’t go back. There is no similarities between cipher & plain text.

Cryptanalysis: science of studying and breaking the encryption processes, authentications schemes, …

Frequency analysis : part of cryptanalysis.

* Based on the frequency of the usage of the letters in the different languages.
* Look for most frequently repeated pattern

Governmental involvement :

* < 1990 : banks & military
* 1990 > : growth of the internet ⇒ several new laws.
  + Key <= 128 bits : easier to decrypted

# Symmetric

* Uses a secret key shared between the receiver & the sender.

⇒ problem : need of lots of keys.

* The security depends on how well users protect the key
* Problem in sharing the keys

Strengths :

* Much faster
* Hard to break if use a large key

Weakness :

* Requires a secure mechanism to deliver keys
* Each pair of users needs a unique key in order to avoid the listening of the communication
* Confidentiality but no authentication

Examples : DES, IDEA, RC5, AES (Rijndael), Twofish

DES :

* First commercial algorithm
* Used from 70’s - 90’s
* Was broken in 3 days using a brute force attack against the keyspace in 1998

AES :

* Block cypher
* Key is doubled or quartered. It takes from 1 sec to 10^12 years → more time to break it (wow, really?)

# Asymmetric

* Uses a set of keys : public and private ones. You always need two keys each time you’re willing to communicate.
* The two keys are mathematically related
* If one key encrypt the message, the other key is required in order to decrypt the message
* Public key can be know to everyone
* Private key must be known only by the owner
* **If confidentiality is most important for a sender, he encrypts the file with the receiver’s public key**
* **If authentication is most important for a sender, he encrypt the file with his private key**

Strengths:

* Better key distribution
* Can provide authentication and nonrepudiation

Weaknesses :

* Works much more slowly than symetric systems
* Mathematically intensive tasks

Examples: RSA, Diffie-Hellman, ECDH, DSA

Diffie-Hellman :

* Sharing 1 key between different entities (a shared network)
* Based on the difficulty to calculate discrete logarithms.
  + Prob : man in the middle ⇒ need a signature so you can authenticate who is who

RSA :

* Most popular public key algo
* Used for digital signatures, key exchange, encryption

### Hybrid cryptography

* Uses symmetric and asymmetric together
* Symmetric algorithm creates keys used for encrypting bulk data
* Asymmetric algorithm creates keys used for automated key distribution
* Use the faster algorithm on the message (symmetric)
* Use the slower algorithm on the key (asymmetric)

### Message integrity :

* One way hash : takes a variable length string ⇒ fixed length value (hash value)
* To be ensure a message does not get altered, calculate a hash value for the message and append it to the message itself.
* Hash message authentication code (HMAC)

### Digital Signature

* Signing means encrypting the message’s hash with a private key
* The hashing provides integrity, the signing can be authenticated by using the public key of the sender, provides non repudiation

### Digital Certificate

Associate a public key with an entity

Standard is X.509

Public key infrastructure (PKI)

* Create a public key ⇒ send it to the RA who checks your identity ⇒ send it to the CA who creates a certificate with a signature ⇒ send it back to the user
* Web servers generates the pair of keys ⇒ public key needs to be registered (get a certificate at the end)
* When connect send the public key and the certificate to prove that you have the correct public in order to send encrypted information

CA are registered on the browser by default for the well-know ones.

### HTTPS

HTTP running over TLS, which uses public key encryption

* Client and server try to use the most powerful encryption protocol and decrease until finding a protocol common to both