Digital signature

Goal is to ensure that

- 1) message is unchanged, 2) the sender's identity is provably genuine
- Any public key cipher combined with hash function can be used in digital signatures
- There exists also special digital signature algorithms, like DSA
- Typical combination is RSA and SHA256

Principle of Digital Signature

CA= key server

Sender's public key

Sender's
public key d

sign

Signature S

verification

Yes/
No

Hash value

Hash

message =>

hash

hash

Sender

Recipient

Digital signature steps

1. Alice calculates the hash value S of the message m
2. Alice encrypts S with her private key d. The result is the signature S’
3. Alice sends the pair (m, S’) to Bob
4. Bob asks Alice’s public key from CA server
5. Bob decrypts the signature with Alice’s public key
6. Bob calculate also the hash of the message received
7. Bob compares the hash values (results of step 5 and 6). If they match, signature is accepted and the message is provably unchanged and sent by Alice.

RSA Digital Signature example

Alice's keys: n = 5 525 310 089 609 , e = 59 627 , d = 3 406 253 797 031

Alice sends message **"Hello, today is November. We are in classroom B310" .** She calculates and sends also the digital signature

Hash of the message h(m) = h = 1574561660

Signature $S = h^{d} \bmod n = 1574561660^{3406253797031} \bmod 5525310089609$

= 3872718136742

Bob receives:

"Hello, today is November. We are in classroom B310" , 3872718136742

Bob calculates the hash of the message from the message part

h("Hello, today is November. We are in classroom B310")= 1574561660

and decrypts the signature part with Alice's public keys:

$3872718136742^{59\,627} \bmod 5525310089609 = 1574561660$

The numbers match => sender is authenticated and message unchanged