Digital Signatures

BMEVITMAV52

Information and Network Security

feher.gabor@tmit.bme.hu

Digital signature

- Message authentication
  - Check whether the message is authentic
  - Check whether the sender is authentic

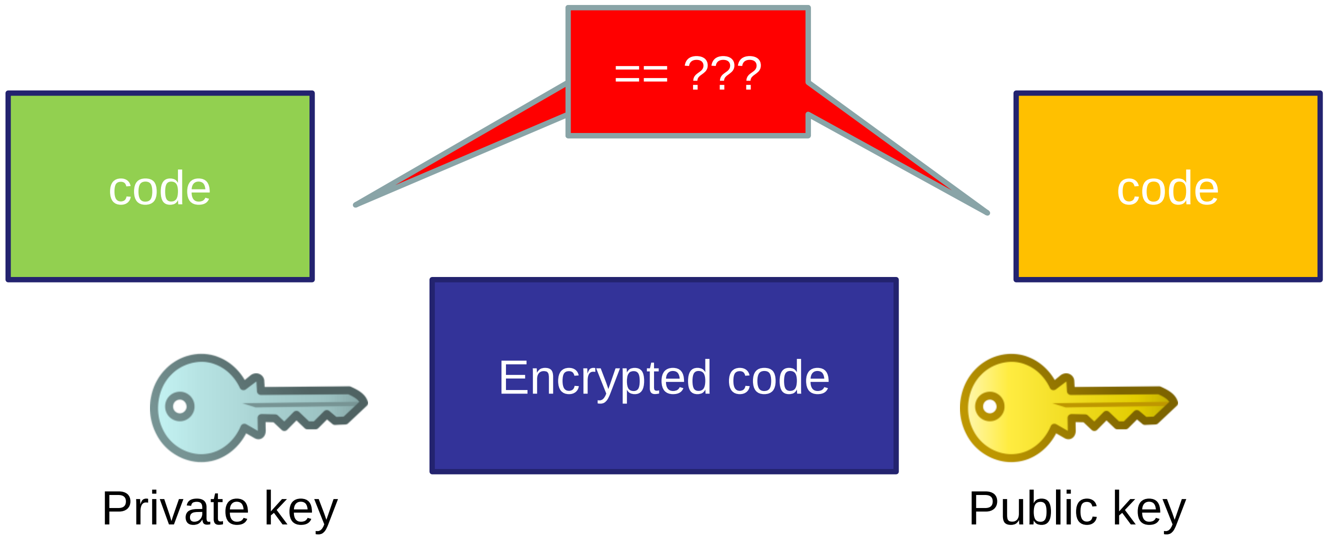
- Based on public-key cryptography
  - The sender's private key is used to create the signature
  - The sender's public key is used to verify the signature

Only the sender knows the private key

Everybody knows the public key

Digital signature

- Successful decryption with the public key means that the encryption was done with the private key
  - Authenticate the key (sender)
  - Authenticate the decrypted text



Hash and sign

- Since public-key encryption is slow, not the whole message, but only its representative image (the hash code) is authenticated
  - Due to hash collisions the message can be fraudulent
    - Due to the birthday attack the complexity is only $2^{n/2}$
- Transfer the message + the signature

RSA signature scheme

- 1. Create public and private keys: e, d, n
  - $n = p \cdot q$, $\varphi = (p-1)(q-1)$
  - $1 < e < \varphi$, $\gcd(e, \varphi) = 1$
  - $1 < d < \varphi$, $e \cdot d \equiv 1 \bmod \varphi$
- 2. Signature generation
  - $m = \text{hash(Message)}$, $0 < m < n-1$
  - Signature: $s = m^d \bmod n$
- 3. Signature verification
  - $m = s^e \bmod n$
  - If $m = \text{hash(Message)}$ then signature is valid

ElGamal signature algorithm

- Create public and private keys
  - p prime, g is generator
  - Select random a: $1 \leq a \leq p-2$, and calculate $A = g^a \bmod p$
  - Public key: p,g,A – Private key: p,g,a
- Signature generation
  - Select random k, $1 \leq k \leq p-2$ and $\gcd(k,p-1)=1$
  - Calculate R: $R = g^k \bmod p$
  - Calculate S: $S = (k^{-1}(h(m)-aR)) \bmod (p-1)$
    - h is a hash function
    - If S would be 0 then repeat with an other k
- Signature verification
  - $g^{H(m)} \equiv A^R R^S \pmod p$

- Proof:
  - $h(m) \equiv aR + kS \pmod{p-1}$, using Fermat's little theorem:
  - $g^{h(M)} \equiv g^{aR} g^{kS} \equiv A^R R^S \pmod p$

- p, g parameters can be shared in the system

Digital signature algorithm (DSA)

- Similar to ElGamal
  - The hash function is the SHA-1
  - Use 160 and1024 bit primes

- Public and private key generation
  - q is a 160 bit prime, p is 1024 bit prime and $p=q \cdot z+1$ for some z
  - Choose h, where $1 < h < p-1$ such that $g=h^z \bmod p > 1$, choose random a
  - $A = g^a \bmod p$
  - Public key: p, q, g, A and private key: p, q, g, a
- Signature generation:
  - Choose random k, $1< k < q$
  - $R = (g^k \bmod p) \bmod q$
  - $S = (k^{-1}(SHA1(m)+ar) \bmod q$
- Signature verification
  - Calculate $w = S^{-1} \bmod q$
  - $R' \equiv (g^{(SHA1(m) \cdot w) \bmod q} \cdot A^{(R \cdot w) \bmod q}) \bmod p \pmod q$
  - Valid if $R' = R$

- p, q, g parameters can be shared in the system

References

- Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, ISBN: 0-8493-8523-7
  - http://www.cacr.math.uwaterloo.ca/hac/
- Wikipedia - The free encyclopedia
  - http://www.wikipedia.org/