

11.1. El Gamal Signature Scheme

Parameters:

p : prime, a : $PE \text{ mod } p$, h : hash functions

Select random x , $y = a^x \text{ mod } p$

Public key: (p, a, y) , Private key: x

Signature generation on a message m :

Compute $h(m)$

Select random k s.t. $k^{-1} \text{ (mod } p-1) \exists$

$$r = a^k \text{ mod } p$$

$$s = k^{-1} (h(m) - xr) \text{ mod } (p-1) \quad (*)$$

Signature on m : (r, s)

Signature verification

Verify: $1 \leq r \leq p-1$

$$v_1 = y^r r^s \text{ mod } p$$

$$v_2 = a^{h(m)} \text{ mod } p$$

$v_1 = v_2 \rightarrow$ accept signature. \perp

Verification works:

$$ks \equiv h(m) - xr \text{ (mod } p-1)$$

$$\Leftrightarrow h(m) \equiv xr + ks \text{ (mod } p-1)$$

$$\Leftrightarrow xr + ks = l(p-1) + h(m) \text{ for some } l \in \mathbb{Z}$$

Hence

$$\begin{aligned} y^r r^s &\equiv a^{xr} a^{ks} \\ &\equiv a^{xr+ks} \equiv \underbrace{(a^{p-1})^e}_{\equiv 1 \pmod{p}} a^{h(m)} \text{ 'Fermat'} \\ &\equiv a^{h(m)} \pmod{p}. \end{aligned}$$

Security

a) Don't use the same key twice! Otherwise

$$s_1 = k^{-1} (h(m_1) - xr) \pmod{p-1}$$

$$s_2 = k^{-1} (h(m_2) - xr) \pmod{p-1}$$

$$\Rightarrow (s_1 - s_2) k \equiv (h(m_1) - h(m_2)) \pmod{p-1}$$

$$\Rightarrow k \equiv (s_1 - s_2)^{-1} (h(m_1) - h(m_2)) \pmod{p-1}$$

provided $(s_1 - s_2)^{-1} \pmod{p-1}$ exists.

Once k is known, x can be obtained from (*)

b) O/E can forge a signature on a message m as follows:

Select any pair (u, v) such that $\gcd(v, p-1) = 1$

Compute $r = a^u y^v = a^u a^{xv} \pmod{p}$

$$s = -rv^{-1} \pmod{p-1}$$

Then (r, s) is a valid signature on

$$m = su \pmod{p-1}. \quad \text{Ex}$$

Avoid this by using hash functions.

$$m \leftarrow h(m).$$

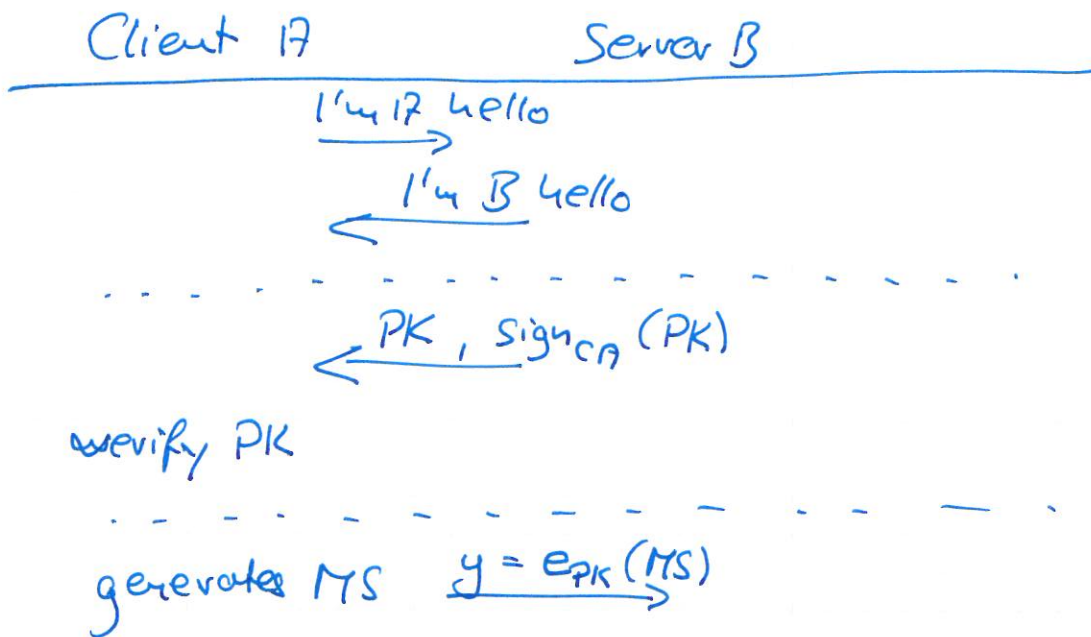
- c) Verification requires checking of $1 \leq r \leq p-1$.
If this omitted then \mathcal{O} can sign
messages of his choice provided he has
one valid signature.

8.4. Public Key Infrastructure (PKI)

Biggest challenge for PKI:
authenticity of public keys.

Examples (TLS/SSL)

Alice wants to buy something from Bob.



$$(K_1, K_2) = h(MS)$$

$$MS = d_{PK}(y)$$

$$(K_1, K_2) = h(MS)$$

K_1 is used to authenticate data by a MAC(K_1).

K_2 is used for en-/decryption (e.g. AES)

Note: A may not have a public key or certificate. B relies on the validity of the credit card number.

—|