

Security

a) Never use the same k twice. Otherwise

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

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

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

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

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

Once k is known, x can be computed from (k) .

b) \odot can forge a signature on a message m as follows.

Select any pair (u, v) s.t. $\text{gcd}(v, p-1) = 1$

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

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

Then (r, s) is a valid signature on

$$m = sr \pmod{p-1}$$

Proof. \odot

Avoid this attack by using hash fct., $h(m)$ instead of m .

c) Verification step requiring $1 \leq r \leq p-1$.

If this check is omitted \mathcal{O} can sign messages of his choice provided he has one valid signature.

Suppose (r, s) is a valid sign. on message m .

\mathcal{O} selects message ~~and~~ m' and computes

$$h(m') \text{ and } u = h(m') (h(m))^{-1} \pmod{p-1}$$

provided $(h(m))^{-1} \pmod{p-1}$ exists.

Further

$$s' = s u \pmod{p-1}$$

$$r' \text{ such that } r' \equiv r u \pmod{p-1}$$

$$\text{and } r' \equiv r \pmod{p}$$

(by the CRT)

The pair (r', s') is a signature on m' , which would be accepted without checking $1 \leq r' \leq p-1$.

Ex

8.4. Public Key Infrastructure (PKI)

Most important components:

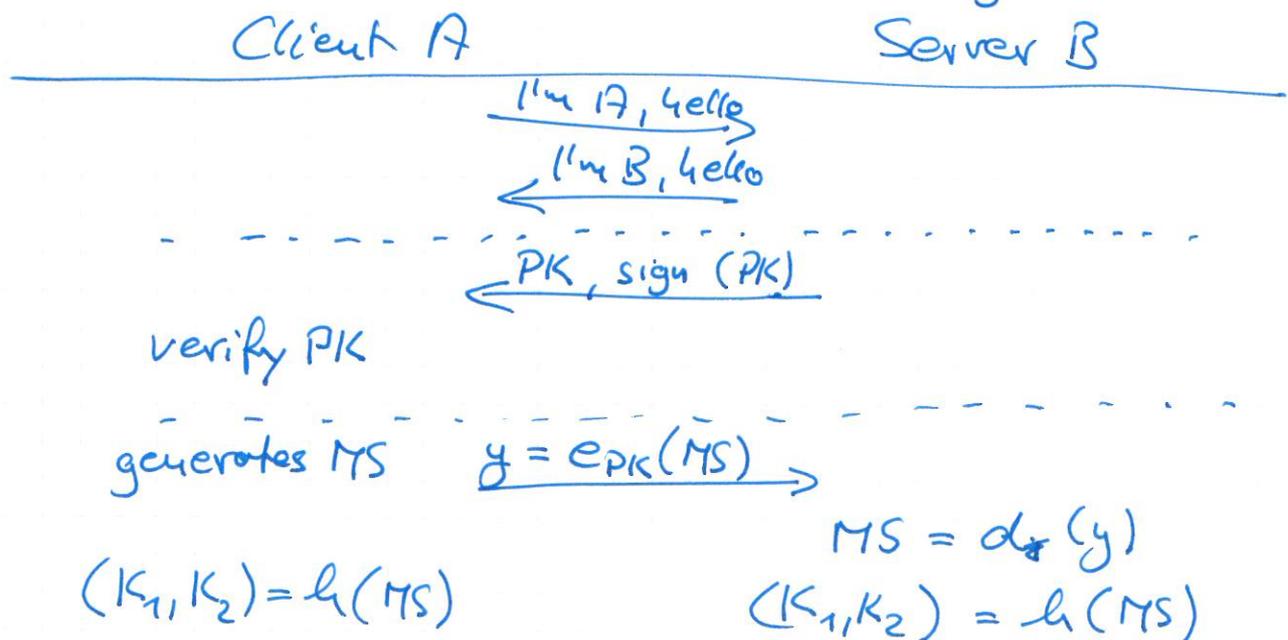
- Certificate issuance
- Certificate revocation
- Key backup/recovery/update
- Time stamping

PKI enables:

- Secure communication
- Access control
- Privacy architecture

Example: SSL (secure socket layer)

A (client) wants to purchase something from B (server).



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

K_2 is used for en-/decryption

(e.g. DES, triple DES, AES, others)

Note: A may not even have a public.

Needed in e-commerce: not the identity of A, but the verification of the credit card no.