C.c) DSA and Diffie-Hellman

C.73 DSA (Digital Signature Algorithm)

- standardized by NIST

A) Generation of a key pair

- Select a prime q with $2^{159} < q < 2^{160}$
- Select a prime p with $q \mid p-1$ and $2^{1023} < p < 2^{1024}$
- Select a generator α of Z_p^* (i.e., $<\alpha> = Z_p^*$)
- Set $g := \alpha^{(p-1)/q}$ (in particular, $|<g>| = q$)
- Select a random number $x \in \{1,\dots,q-1\}$
- $y := g^x \pmod p$

Secret key: x

Public Key: (y,p,q,g)

C.73 (continued)

B) Generation of a digital signature

- generate a random number $k \in \{1,\ldots,q-1\}$ (ephemeral key)
- $r := (g^k \pmod p) \pmod q$
- $s := k^{-1}(H(m)+xr) \pmod q$

H denotes a hash function. In the DSS (Digital Signature Standard) H=SHA-1.

C.73 (continued)

C) Verification of a digital signature

- verify that $0 < r,s < q$
- $u_1 := s^{-1} H(m) \pmod q$
- $u_2 := s^{-1} r \pmod q$
- $v := (g^{u\_1} y^{u\_2} \pmod p) \pmod q \stackrel{?}{=} r$

Justification:

$g^{u\_1} y^{u\_2} \equiv g^{s\text{^}(-1)H(m)} g^{xs\text{^}(-1)r} \equiv g^{s\text{^}(-1)(H(m)+xr)} \equiv g^k \pmod p$

C.74 DSA (Security)

- The security of DSA essentially grounds on the discrete log problem in the subgroup $< g > \subseteq Z_p^*$ (recall that $y := g^x \pmod p$).
- Unlike RSA the DSA algorithm needs a fresh random number k (ephemeral key) for each signature. In particular, if Alice signs the same message m several times all signatures will be different.
- If an attacker knows k it is easy to solve the linear equation $s := k^{-1}(H(m)+xr) \pmod q$ over the field GF(q) to determine the secret key x.
- Applying lattice-based attacks it is sufficient if an attacker knows small parts of the ephemeral keys from a large number of signatures.

C.75 DSA (Efficiency)

- Since k is only a 160 bit integer the signature generation is much faster than for 1024-bit RSA, for instance. Moreover, the value r may be precomputed.
- The signature verification is significantly more costly than for RSA signatures with small public exponents.

Note: DSA can only be used for signing, not for encryption (key exchange).

C.76 Diffie Hellman Key Agreement Protocol (Basic Variant)

- Goal: Alice and Bob want to agree upon a secret key. An adversary shall not be able to recover this key.

First Step: Alice and Bob agree upon a prime p, a generator $g \in Z_p^*$ (or at least on an element with large order) and a key derivation function f. These parameters may be made public.

C.76 (continued)

- Alice selects randomly $a \in \{1,\ldots,p-2\}$ and keeps this value secret.
- Bob selects randomly $b \in \{1,\ldots,p-2\}$ and keeps this value secret.
- Alice sends $A:=g^a \pmod p$
- Bob sends $B:=g^b \pmod p$
- Alice computes $C:=B^a \equiv g^{ab} \pmod p$ and $k=f(C)$
- Bob computes $C:=A^b \equiv g^{ab} \pmod p$ and $k=f(C)$

<u>Note:</u> Alice and Bob have agreed upon the key k.

C.77 Remark

- The basic version of Diffie-Hellman’s key agreement protocol is vulnerable against active adversaries. An active adversary could e.g. send any value $E:=g^e \pmod p$ to Bob, pretending being Alice.
- Hence the basic protocol is embedded into more advanced protocols.
- The underlying idea can also be used to encrypt messages (cf. e.g. the ElGamal encryption scheme).

C.78 Elliptic Curve Cryptography

- Key agreement protocols and signature applications that are based on elliptic curves have become increasingly important. Compared to RSA shorter key lengths provide a similar security level ($\rightarrow$ efficiency).
- Elliptic curve-based cryptographic algorithms are more difficult to understand than RSA. Elliptic curves are beyond the scope of this course.
- We just mention that elliptic curves over finite fields are finite abelian groups. For suitably selected parameters the discrete log problem on elliptic curves is intractable.
- In particular, there exists a pendant to the DSA algorithm (ECDSA).

C.79 Final Remark

- In this course we merely scratched the field of public key cryptography.
- There exist several other mechanisms and protocols that we have not even addressed, e.g. blind signatures (discussed in the exercises) and zero-knowledge proofs.