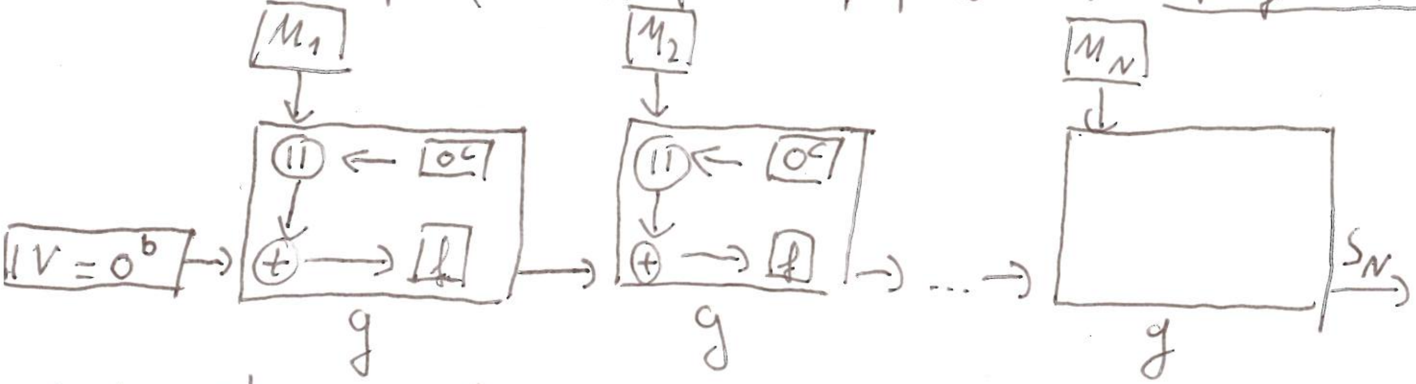
Pocket calculator list for the exam is to be found in L2P in the section Hyperlinks

Construction principle for hash functions, special case: <u>Sponge construction</u>

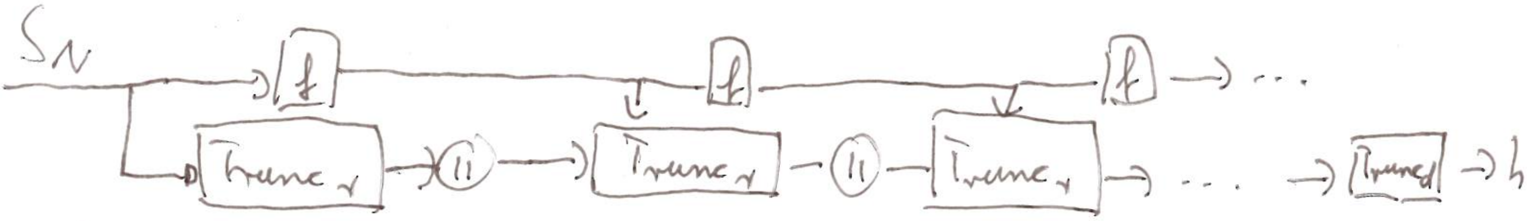


$f: \{0,1\}^b \to \{0,1\}^b$

$b = r + c$ $\quad r > 0$ rate $\quad c$: capacity; $M_i \in \{0,1\}^r$; N msg blocks

Generating is done in the so called absorbing phase, where the message is generated.

In the so called squeezing phase the output of length d is generated as follows



where h is the final hash value of length d

c) Verification of step (i) of El Gamal signatures requires checking of $1 \leq r \leq p-1$

If this check is omitted then Oscar can sign messages of his choice provided he has one valid signature and $h(m)^{-1} \bmod p-1$ should exist.

Suppose (r, s) is a signature for message m.

O selects a message m' of his choice and computes $h(m')$ and $u = h(m')\,(h(m))^{-1} \bmod p-1$

He defines $s' = s \cdot u \bmod p-1$

Then there exists a pair (r', s') which is a signature for m' which would be accepted, if $1 \leq r' \leq p-1$ is ignored.

11.2 The Digital Signature Algorithm (DSA)

- Proposal by the NIST in Aug '91
- Standardized as FIPS 186, named DSS (Digital Signature Standard)
- Developed by the NSA (not publicly)
- DSA is a variant of the ElGamal signature scheme
- Needs a hash function $h: \{0,1\}^* \to \mathbb{Z}_q$ as a building block
  The standard prescribes SHA-1.

System parameters

Each user generates a public and private key as follows:

1. Choose a prime q with $2^{159} < q < 2^{160}$ (160 bits)
2. Choose t, $0 \leq t \leq 8$, further a prime p such that

   $2^{511+64t} < p < 2^{512+64t}$ and $q \mid p-1$ (512 .. 1024 bits)

   Recommended by NIST from Oct. 2001: $t = 8$, i.e., 1024 bits
3. (i) Select a PE $g \in \mathbb{Z}_p^*$, compute $a = g^{(p-1)/q} \mod p$

   (ii) If $a = 1$, repeat step (i)

   (a is a generator of a cyclic subgroup of order q in $\mathbb{Z}_p^*$)
4. Choose random $x \in \{1, \ldots, q-1\}$
5. Compute $y = a^x \mod p$
6. Public key: (p, q, a, y), private key x

Signing a message $m \in \{0,1\}^*$

1. Choose a random $k \in \{1, \ldots, q-1\}$
2. $r = (a^k \mod p) \mod q$
3. Compute $k^{-1} \mod q$
4. $s = k^{-1}(h(m) + x \cdot r) \mod q$
5. signature (r, s) (320 bits in total)

Verification of signature (r,s) on message m:

1. Check if $0 < r < q$ and $0 < s < q$, otherwise decline
2. $w = s^{-1} \bmod q$
3. $u_1 = (w \cdot h(m)) \bmod q$, $u_2 = (r \cdot w) \bmod q$
4. $v = (a^{u_1} \cdot y^{u_2} \bmod p) \bmod q$
5. Accept the signature if $v = r$

Proof that the verification is correct:

For a valid signature (r,s) it holds that

$$h(m) \equiv k \cdot s - x \cdot r \pmod q$$

Hence, $a^{u_1} \cdot y^{u_2} \equiv a^{u_1} \cdot a^{x u_2} \pmod p$

$$u_1 + x u_2 \equiv w \cdot h(m) + x \cdot r \cdot w \equiv w(k \cdot s - x \cdot r) + x \cdot r \cdot w \pmod q$$

$$\equiv w \cdot k \cdot s \overset{2.}{\equiv} k \pmod q$$

because a has order q

$$v \equiv (a^{l \cdot q + k} \bmod p) \bmod q \equiv (a^k \bmod p) \bmod q = r$$

Security

- Security relies on two DL problems
  a) in $\mathbb{Z}_p^*$ b) in $\langle a \rangle \leq \mathbb{Z}_p^*$ $\langle a \rangle$ denotes the subgroup generated by a
- Security principle of the ElGamal scheme carry over:
  - Always choose a new k
  - Use of a hash function is mandatory
  - Always verify 1. in the verification process. Otherwise signatures for arbitrary messages can be generated provided one valid signature is known.

Remarks /

a) Modular exponentiation is in the range of q (160 bits) (rather than 1024 El Gamal)

b) $k, k^{-1}, r, x r$ may be generated, computed and stored in advance

c) Verification needs 2 instead of 3 modular exponentiations

d) Signature by DSA is short, 320 bits, instead of 2048 bits for El Gamal.

e) In the verification step, also check if $r \neq 0$, $s \neq 0$ otherwise the signature is rejected. But this happens with a very small probability

12. Identification and Entity Authentication

This chapter considers techniques to allow the "verifier" to establish the identity of the "claimant", thereby preventing impersonation.

Requirements on authentication protocols:

1. A is able to uniquely identify herself to B
2. B cannot reuse an identification exchange with her so as to impersonate A to a third party C. (transferability)
3. It is practically infeasible that a third party C can cause B to wrongly accept the identity of A (impersonation)
4. Even if C observes the identification process between A and B very often he cannot impersonate A.

Three main categories of identification:

1. Something is known: password, PIN, private key
2. Something possessed: key, magnetic-striped cards, chipcards, PIN or password generator
3. Something inherent: human physical characteristics, face recognition, fingerprint, retinal pattern, handwritten signature

12.1 Passwords

Fixed password schemes

Rather than storing a cleartext user password (pwd) in a file, a hash value h(pwd) of each user password is stored. Verification is done by comparing the hash value of the entered password with the stored one for a given user.

Main attacks are

- replay of fixed password
- exhaustive password search
- password - guessing and dictionary attack

Defense strategies are

- Choose a random password, or nearly random, use of special characters (increasing entropy)
- Slowing down the password mapping
- Salting passwords

Extend the password by some random string, the salt, before hashing. Both the hashed password and the salt are stored

h(password, salt), salt

This does not complicate exhaustive search, but, simultaneously dictionary attacks a large set of passwords.

One-time passwords

Protect against eavesdropping and replay of passwords or "phishing".