Crypto 2 Ü 10

E29.) a) Suggest a probabilistic algorithm to find a pair of primes
(t = 8 chosen, NIST recommended) p, q with

i.) $2^{159} < q < 2^{160}$
ii.) $2^{1023} < p < 2^{1024}$
iii.) $q \mid p-1$

Idea: 1.) Get a random odd number q with $2^{159} < q < 2^{160}$
2.) Construct p such that i.) and iii.) are fulfilled.
Repeat, if p not prime

iii.) $\Leftrightarrow \exists k \in \mathbb{N} \quad k \cdot q = p-1$
$\Rightarrow k$ even as q odd and $p-1$ even

$$p = k \cdot q + 1 \overset{\text{i.)}}{>} 2^{1023} \Leftrightarrow k > \frac{2^{1023}-1}{q}$$

$$p = kq + 1 \overset{\text{ii.)}}{<} 2^{1024} \Leftrightarrow k < \frac{2^{1024}-1}{q}$$

1.) Get a random odd number q with
$2^{159} < q < 2^{160}$
Repeat if not prime

2.) Get a random even number k with
$$\frac{2^{1023}-1}{q} < k < \frac{2^{1024}-1}{q}$$
If $p = k \cdot q + 1$ is not prime repeat 2

b.)

For testing if q and p are prime in 1.) and 3.) a primality test chosen, e.g. MRPT such that the error probability is neglectible. The success probability of finding a prime of size x is about $\frac{1}{\ln(x)}$

If you skip even numbers (which are not prime) you double the success probability

Taking the hint into account leads to

$$p = \frac{2}{\ln(2^{160})} \cdot \frac{2}{\ln(2^{1024})} = \frac{1}{80.512 \cdot \ln(2)^2} \approx 5{,}08 \cdot 10^{-5}$$

Ex 30)

DSA Public key (p, q, a, y)

private key (x) $(y = a^x \bmod p)$

Modified verification without hash function (see 11.2.)

1.) check, if $0 < r < q$ and $0 < s < q$

2.) $w = s^{-1} \bmod q$ [and $h(m)$] orig. alg.

3.) Compute $u_1 = w \cdot m \bmod q$

$u_2 = r \cdot b \bmod q$

Difference to orig. alg

4.) Compute $v = (a^{u_1} \cdot y^{u_2} \bmod p) \bmod q$

5.) Accept, if $v = r$

Hint: see 11.1.1.-2:

choose pair (u, v) such that $\gcd(v, q) = 1$

Calculate $r = (a^u y^v \bmod p) \bmod q$

$s = r \cdot v^{-1} \bmod q$ $[-r \cdot v^{-1} \bmod p-1]$

(derived on next page)

Claim: Then (r,s) is a valid signature for the message $m \equiv s \cdot u \mod q$

1.) ✓

2.) Later

3.) $u_1 = w \cdot m \equiv s^{-1} \cdot m \equiv s^{-1} \cdot s \cdot u \overset{!}{\equiv} u \pmod q$

$u_2 = r \cdot b \equiv r \cdot s^{-1} \overset{!}{\equiv} v \pmod q$

$\Leftrightarrow s \equiv r \cdot v^{-1} \pmod q$

4.) $a^{u_1} y^{u_2} \overset{q \mid p-1}{\equiv} a^u y^v \pmod p$

5.) $\Rightarrow v = r$

Ex 31.)

a.) (A, i, w_i) known from previous transmission

How to compute $(A, i+1, w_{i+1})$

$w_{i+1} = H^{t-(i+1)}(w) = H^{-1}(H^{t-i}(w)) = H^{-1}(w_i)$

does not exist!

As H is one-way, this is not possible even if H is known

b.) H has to be preimage resistant

Given $y \in Y$ it is infeasible to find m such that $H(m) = y$

otherwise $w_i = H(w_{i+1})$ would be broken see a.)

c.) DLP (discrete log problem) is hard in $\mathbb{Z}_p^*$,
i.e., H is hard to find x with $a^x \equiv y \bmod p$

Use: $H: \{2, \dots, p-2\}$ (secret) $\to \mathbb{Z}_p^*$,

$w \mapsto a^w \bmod p$

- choose $w \in \{2, \dots, p-2\}$ (secret)
with $a^w \neq 1 \bmod p$
- $w_0 = H^t(w)$
- Protocol: $H^{t-i}(w) = w_i$

$A \to B$ ~~(w_{i-1})~~ (A, i, w_i)

B checks if $w_{i-1} = a^{w_i} \bmod p$

d.) Man-in-the-middle-attack

Adversary impersonates B, gets PW from A,
can use this for authentication to B.