# Cryptography, winter 2016/17

MICHAEL NÜSKEN, JAKOB NUSSBAUMER

## 10. Exercise sheet
## Hand in solutions until Friday, 20 January 2017, 12:00 (noon)

**Exercise 10.1** (Cleptography). (12+3 points)

Someone has replaced the key generation with the following:

**Modified Key Generation.**
Input: Security parameter $\kappa \in 8\mathbb{N}$.
Output: Public key $[N, e]$, $[N, e]$, private key $[N, d]$.

1. Fix a pseudo random generator with seed $s$ for use wherever the subsequent algorithm uses random bits.
2. Pick $p, q_0 \xleftarrow{\text{RR}} \mathbb{P}$ with $2^{\kappa-1} \leq p \cdot q_0 < 2^{\kappa}$ and...
3. Compute $N_0 \leftarrow p \cdot q_0$.
4. Compute an encryption of $s$ for Eve with any encryption algorithm (for example a one-time pad), just ensuring that $\mathrm{Enc}_{\mathrm{Eve}}(s) \in \{0,1\}^{\frac{\kappa}{4}}$.
5. Take the bit pattern of $N_0$ and replace the bits $\frac{7\kappa}{8} - 1 \ldots \frac{5\kappa}{8}$ with $\mathrm{Enc}_{\mathrm{Eve}}(s)$. The result is called $N_1$.
6. Compute $q := \text{nextprime}\left(\frac{N_1}{p}\right)$.
7. Compute $N \leftarrow p \cdot q$.
8. Compute $L \leftarrow (p-1) \cdot (q-1)$.
9. Pick $e, d \in \mathbb{N}$ with $e \cdot d = 1$ in $\mathbb{Z}_L$.
10. Replace this algorithm in the library by the original one.
11. Return $[N, e]$, $[N, d]$.

We may assume $q \leq \frac{N_1}{p} + c\kappa$ for some small $c$ with high probability. This is reasonable.

(i) Prove that $0 < N - N_1 < 2^{\frac{\kappa}{2} + \log_2 \kappa + \log_2 c + 7}$. $\boxed{2}$

Assume that $\log_2 \kappa + \log_2 c + 7 < \frac{\kappa}{8}$.

(ii) Conclude that $N$ and $N_1$ coincide in their bits $\frac{7\kappa}{8} - 1 \ldots \frac{5\kappa}{8}$ assuming that no carry to bit $\frac{5\kappa}{8}$ occurs when adding $N_1$ and $N - N_1$. $\boxed{2}$

(iii) Explain what Eve can do based on the public key. $\boxed{3}$

(iv) Can this modification be detected? $\boxed{2}$

(v) Conclude that it is vital to somehow document/prove the provenience of the key pair. $\boxed{3}$

Draw consequences. (Behaviour? NIZK?) $\boxed{+3}$

**Exercise 10.2** (Primality Testing). (10+10 points)

In this exercise we put hands on the primality tests discussed in the lecture.

3 (i) Implement the Miller Rabin test in a programming language of your choice.

*Hint*: At `http://www.sagemath.org/` you can use SageMath online or download it (native on Linux, or in a VirtualBox on Windows) for local use. Sage is python-based.

3 (ii) Implement the Fermat test in a programming language of your choice, ie. take the Miller Rabin test and omit the line leading to the answer: "composite" (with factor).

Now, let's run it! Execute the Miller Rabin test with

1 (iii) $N = 41$, $x = 2$.

1 (iv) $N = 57$, $x = 37$.

1 (v) $N = 1105$, $x = 47$.

1 (vi) $N = 1105$, $x = 2$.

With our implementation running, we can now perform several experiments.

+2 (vii) Compute the number of Fermat liars for $N = 35$, ie. the number of choices $x \in \mathbb{Z}_N$ for which the Fermat test returns $N$ "maybe prime".

+1 (viii) Compute the number of Miller Rabin liars for $N = 35$, ie. the number of choices $x \in \mathbb{Z}_N$ for which the Miller Rabin test returns $N$ "maybe prime".

+1 (ix) Do the same for $N = 561$.

+6 (x) Perform more experiments and interpret the results.