# Cryptography, winter 2013/2014

PROF. DR. JOACHIM VON ZUR GATHEN, DR. DANIEL LOEBENBERGER

## 5. Exercise sheet
## Hand in solutions until Saturday, 30 November 2013, 23:59:59

**Exercise 5.1** (Reductions for RSA). (7+1030 points)

We consider as an attacker a (probabilistic) polynomial-time computer $\mathcal{A}$. $\mathcal{A}$ knows pk $= (N, e)$ and $y = \text{enc}_{\text{pk}(x)}$. There are several notions of "breaking RSA". $\mathcal{A}$ might be able to compute from its knowledge one of the following data.

$B_1$: the plaintext $x$,

$B_2$: the hidden part $d$ of the secret key sk $= (N, d)$,

$B_3$: the value $\varphi(N)$ of Euler's totient function,

$B_4$: a factor $p$ (and $q$) of $N$.

If $A$ and $B$ are two computational problems (given by an input/output specification), then a *random polynomial-time reduction* from $A$ to $B$ is a random polynomial-time algorithm for $A$ which is allowed to make calls to an (unspecified) subroutine for $B$. The cost of such a call is the combined input plus output length in the call. If such a reduction exists, we write

$$A \leq_p B.$$

(i) Show that $B_1 \leq_p B_2$. $\boxed{2}$

(ii) Show that $B_2 \leq_p B_3$. $\boxed{2}$

(iii) Show that $B_3 \leq_p B_4$. $\boxed{2}$

(iv) Which problem is the easiest one? Which one is most difficult? $\boxed{1}$

(v) Show that additionally we have $B_4 \leq_p B_3$. Hint: Consider the quadratic polynomial $(x - p)(x - q) \in \mathbb{Z}[x]$. $\boxed{+2}$

(vi) Argue that we also have $B_3 \leq_p B_2$. $\boxed{+4}$

(vii) Resolve the question whether also $B_2 \leq_p B_1$ or equivalently whether $B_4 \leq_p B_1$. Warning: This is an open research problem... $\boxed{+1024}$

**Exercise 5.2** (RSA bad choice). (6 points)

Show why the 35-bit integer 23360947609 is a particularly bad choice for $N = pq$.

We claim that two prime numbers which are really close to each other are bad choices for RSA system. To show this we use Fermat's factorization method based on the fact: If $N = pq$ with $p > q$ being odd primes, then $N = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$.

4 (i) Explain how you can use this fact to find prime factors of $N$.

2 (ii) Do it for $N = 23360947609$.

**Exercise 5.3** (Primality Testing). (10+10 points)

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

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

3 (ii) Implement the Strong pseudoprimality test in a programming language of your choice.

Now, let's run it! Execute the Strong pseudoprimality 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 $N = 35$, i.e. the number of choices $x \in \mathbb{Z}_N$ for which the Fermat test returns "$N$ is possibly prime".

+2 (viii) Compute the number of Strong liars for $N = 35$, i.e. the number of choices $x \in \mathbb{Z}_N$ for which the Strong primality test returns "$N$ is probably prime".

+2 (ix) Do the same for $N = 561$.

+2 (x) Perform more experiments.

+2 (xi) Interpret the results.