# Cryptography, winter 2016/17

MICHAEL NÜSKEN, JAKOB NUSSBAUMER

## 12. Exercise sheet
## Hand in solutions until Friday, 13 January 2017, 12:00 (noon)

**Exercise 12.1** (Finger exercise). (9+7 points)

Prove some goodies:

(i) RSA is not IND-CPA secure. Generally: each deterministic public-key encryption scheme is not IND-CPA secure. 2+2

(ii) ElGamal encryption scheme is not IND-CCA secure. Generally: each homomorphic public-key encryption scheme is not IND-CCA secure. 2+2

(iii) Where does the previous proof fail for the Cramer & Shoup encryption scheme. +3

(iv) ElGamal encryption scheme is IND-CPA if DDH is hard relative to GenGroup. 5

**Exercise 12.2** (Security reduction). (6 points)

For a signature scheme, a message is first hashed and then the hash value is signed. Assume that the signature scheme is secure in the EUF-CMA model. Does that imply that the hash function is collision resistant? Formulate a precise statement and prove your answer. 6

**Exercise 12.3** (RSA Hardcore Bit). (12+6 points)

In this exercise we will examine the question whether an algorithm that gives you partial information on the plaintext (given the public key and the ciphertext) already gives you the complete plaintext.

(i) First assume that you are given an algorithm BitZero that on input $(N, e, y)$ outputs the least significant bit of the plaintext $x$ (so it says whether $x$ is even or odd). Construct given BitZero an algorithm $\mathcal{A}$ on input $(N, e, y)$ produces the whole plaintext $x$. [Hint: If $\mathcal{A}(N, e, y) = 0$ then $x = 2x'$. Otherwise note that $N$ is odd!] 8

Implement your solution and check it on a random example. 4

(ii) Often one has probabilistic algorithms which will not always give the correct answer, but work with a certain error probability. You are now going to explore how such an algorithm would behave in our setting. So assume now that the algorithm BitZero has a small error probability +3

of $2^{-n}$ where $n$ is the number of bits in $N$. Compute the probability that your algorithm $\mathcal{A}$ returns the correct plaintext. [Hint: The Bernoulli inequality states that $(1+x)^r \geq 1 + rx$ for $x > -1$ and $r \geq 0$.]

+3 (iii) Finally assume that the attacking algorithm has a huge error probability of $40\%$. Can you still compute the entire plaintext efficiently?