# Exercise Sheet

## Information Security II

Summer Term 2018

12 April 2018

**Purpose of this Exercise Sheet** This exercise sheet gives you an idea about the workload of the homework assignment distributed on 17 May. It should not be difficult to come up with a correct solution using the lecture material, the referenced papers, a textbook, or even the information on the English Wikipedia. We will discuss your solutions in the proseminar on 26 April.

**1) Valgrind (3 Points)** Install *valgrind* (e. g., from `http://valgrind.org/`). Then write a simple C program that triggers a warning of the valgrind tool *memcheck*. (Remember the Debian OpenSSL vulnerability as a starting point.)

**2) Hardware in the Hands of the Enemy (1, 2, 3 Points)** Assume that your ski pass performs a Diffie–Hellman key exchange via RFID with the reader of the automatic gate. The ski pass uses the established key to encrypt and then send its ID (a number) to the gate. The gate makes an identity-based authorization decision before it lets you pass. The ID is a pre-shared secret between the skiing resort and the cards it distributes. It is unique for each customer and must not leave the system in order to prevent the cloning of ski passes. Assume that you can buy clean RFID passes from China that support the same protocol and can be programmed at home.

2.a. Is this protocol secure? If not, which protocol-level attack allows you to clone the ski pass.

2.b. How could you improve the protocol?

2.c. Which other attacks are possible against your improved protocol if the attacker can carry some hardware in his backpack?

**3) Defenses Against RSA Fault Injection (2, 3 Points)**

3.a. In the lecture you learned about a defense against random fault injections on RSA decryption with the CRA that decrypts twice and compares the results. Does this idea still promise performance gains compared to the more secure alternative of decrypting RSA without the CRA as $n$ grows very large?

3.b. Now consider deterministic fault injections, e. g., through backdoors in multiplier hardware. Can you come up with a countermeasure that still

- allows you to decrypt every input chosen by the adversary,
- does not limit the word size of the architecture, and
- uses the CRA?

Hint: random numbers are cheap.

**4) PINs, Pollen, Probability (1, 4, 1 Points)** Consider a key pad that gives you access to a building. You know that one 5-digit PIN is used and you can see from the key pad that exactly 5 keys are worn off.

4.a. How many trials do you need on average to open the door without prior knowledge of the PIN?

4.b. It's springtime and pollen are everywhere. How and by how much can you reduce the number of trials if you can selectively wipe keys before a person enters the building and inspect the key pad thereafter? Assume that the person uses only one finger. If this finger has first touched a dusty key, traces of the dust will be visible on all subsequently pressed clean keys.

4.c. How do you defend against this attack

- as operator of the building and
- as the person entering the building?