CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

**Formal Analysis of Real-World Security Protocols**

Prof. Dr. Cas Cremers, Dr. Aleksi Peltonen
Alexander Dax, Niklas Medinger

https://cms.cispa.saarland/farwsp24/

# Exercise Sheet 1

due 04.11.2024 at 14:00

---

## Instructions:

You should submit exercises in groups of **two**. Submissions **must** include the name and matriculation number of **both** students. Please submit only once per group. Solutions must be submitted in PDF format to CMS. You should provide solutions in English. If there are additional files, they can be downloaded from the material section in CMS.

---

## Exercise 1: The Symbolic Model - Usage, Tools and History (30 points)

(a) (3 points) In the lecture, you heard the term *symbolic model*. Define it in your own words and state when and why it was proposed. (max. 3-4 sentences)

(b) (3 points) In contrast to the symbolic model of cryptography, there is also the more commonly known *computational model* of cryptography. You may be familiar with this model from other cryptography lectures. Up to this day, cryptographers use the computational model to prove cryptographic constructions and protocols secure. State three major differences you can find between these models.

(c) (12 points) When using the symbolic model, we often use tools developed to assist our analysis. Find 4 tools that (semi-)automate proving in the symbolic model.

(1) Name them and state when they were published.

(2) For each tool, find 3 security protocols or constructions that were analysed using it.

(d) (6 points) Name 3 major improvements that made the analysis of protocols in the symbolic model more feasible since its proposal.

(e) (6 points) You have decided to analyze the design of a protocol whose specification you recently encountered online. You plan to start modeling the protocol, but a friend advises caution, warning that there are potential pitfalls even when working with a formal specification. Name three common mistakes one can do while modeling a protocol.

## Exercise 2: Equational Theory (35 points)

(a) (16 points) In the lecture, you saw many cryptographic primitives and data structures, and their equational theories; for instance, symmetric encryption, digital signatures or pairs.

For this and the following exercises, assume that natural numbers are defined by:

```
functions: Z/0, S/1
equations:
```

where $Z()$ represents the number 0 (zero) and $S(x)$ represents the successor of $x$, where $x$ is a natural number. For instance, the number 2 would be equal to `S(S(Z()))`.

Give (`functions`, `equations`) for

(1) message authentication codes (MAC)
(2) a randomized version of digital signatures
(3) addition of natural numbers
(4) multiplication of natural numbers

(b) (5 points) Symmetric encryption is usually modeled as follows.

```
functions: senc/2, sdec/2
equations: sdec(senc(m,k),k) = m
```

Now consider encryption modes like ECB[1] or OFB[2]. Is the model described above still a reasonable model? If yes, explain why. If no, give an updated model.

(c) (14 points) Recall the definitions of hash functions and randomized signatures.

(1) Assume that you found a weakness in a randomized signature scheme that allows you to—given a valid signature—construct a forged signature which verifies under a value of your choosing instead of the initial used randomness. How would you change the signature model to capture this behaviour? Give the (`functions`, `equations`).
(2) Oops, you did it again. You broke another cryptographic primitive. You somehow managed to find a way to efficiently invert hash functions. Update the (`functions`, `equations`) accordingly.

## Exercise 3: Term Rewriting (35 points)

Let $\Sigma = \{Z, S, add\}$, where $Z$ is a constant function, $S$ is a function of arity 1, and $add$ is a function of arity 2. Additionally, let $\mathcal{R}$ be the set of rewrite rules over $\Sigma$ with $\mathcal{R}$ containing

$$add(x, Z()) \to x \tag{1}$$
$$add(Z(), y) \to y \tag{2}$$
$$add(S(x), S(y)) \to S(add(x, y)) \tag{3}$$

with $x, y \in \mathcal{T}_\Sigma(\mathcal{V})$

(a) (6 points) Consider the term $t = add(Z(), add(S(add(S(S(Z())), Z())), S(Z())))$. For each position $p$ in $t$, identify and list all subterms $t|_p$, either by listing them individually or by presenting them in a tree structure.

(b) (3 points) The function $add$ is intended to model the addition of natural numbers. The rewrite system $\mathcal{R}$ seems to be faulty as $add$ does not represent standard addition. Propose corrections to the rewrite system $\mathcal{R}$ to fix the faulty behavior.

(c) (6 points) With the corrected rewrite system from (b), apply the rewrite rules exhaustively to the term $t$ from (a). Perform the rewriting step by step. What natural number does the result represent?

(d) (10 points) We now extend $\Sigma$ with two functions $max$ and $min$, both with arity 2. These functions should represent the mathematical minimum and maximum and hence should return the lowest and highest value of two numbers, respectively. Come up with extensions to the rewrite system $\mathcal{R}$ that emulate this behaviour for terms.

(e) (10 points) We now want to include a representation to compute Fibonacci numbers[3]. Add the necessary function symbols to $\Sigma$ and extend $\mathcal{R}$ to include rewrite rules that emulate a function capable of computing the n-th Fibonacci number, given n as input.

[1] `https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Electronic_codebook_(ECB)`
[2] `https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Output_feedback_(OFB)`
[3] `https://en.wikipedia.org/wiki/Fibonacci_sequence`