Problem sheet 4 for Course 02231, 2025

These practice problems have the purpose of helping you understand the material better and learning the skills that are necessary to analyze cryptographic constructions, and sometimes to prepare you for the next class. All answers should be supported by a written justification. To gauge whether a justification is sufficient, ask yourself if your peers would be convinced by it without additional explanations.

We denote vectors as $\mathbf{x} \in \{0,1\}^\lambda$. By $\mathbf{x}[i]$ we denote the ith index of $\mathbf{x}$, where $i \in \{1, \ldots, \lambda\}$. As in the lecture, we write $k \leftarrow K$ if k is sampled from the set K such that it can be each element from K with equal probability $1/|K|$.

❓ **Exercise 1.** (Never use ECB)

In the lecture, it was mentioned that the ECB mode of operation should never be used for anything, under any circumstances. In this exercise we will see that it is not even CPA secure.

1. Assume ECB is used with a block cipher of block length B. Consider the encryptions of the messages $\underbrace{0\ldots0}_{B \text{ bits}}|\underbrace{1\ldots1}_{B \text{ bits}}$ and $\underbrace{0\ldots0}_{B \text{ bits}}|\underbrace{0\ldots0}_{B \text{ bits}}$. Can you see a problem?
2. Turn the discovered problem into an attacker against left-right CPA security. Your attacker should make 1 query to the library, and always succeed.

❓ **Exercise 2.** (Padding for modes of operation)

In the lecture, we saw that modes of operation allow to encrypt messages of length $\ell \cdot B$ bits, where B is the block length and ℓ is an integer. Unfortunately, not every message is an exact multiple of B. To support messages of arbitrary lengh, a so-called *Padding scheme* is used.

The ANSI X.923 padding works as follows:

1. Let m be the message of n bytes length, such that $\ell = \lfloor 8n/B \rfloor$.
2. Output the message $m' = \underbrace{m|00|00|\ldots|00|b}_{(\ell+1)\cdot(B/8) \text{ bytes}}$, where b is the number of bytes that had to be added to m. b itself is encoded in one byte, and therefore has length 8 bits.

For example, if m is 4 bytes short of being a multiple of B, then one would append 00|00|00|04 as padding. If it is just missing two bytes, then we instead add 00|02.

1. Assume that m already has a length that is a multiple of B and that it ends with 01. Why do we have to add a whole block of length B to m before encrypting it, instead of just encrypting the original message?
2. What is the maximal block size (in bits) that can be supported by the ANSI X.923 padding scheme?
3. Consider the CTR mode of operation. It can be modified to be able to encrypt messages whose lengths are not multiples of B. How can this be done?

❓ Exercise 3. (Adversarial Sampling)

In the class, it was claimed that any polynomial-time (in λ) attacker only has negligible (in λ) distinguishing advantage for the following libraries:

Library $L_{\textbf{GenSample}}$:

```
$\underline{\mathit{Sample}(R \subset \{0,1\}^\lambda)}$:
1. $r \leftarrow \{0,1\}^\lambda$
2. Output $r$
```

Library $L_{\textbf{GenSampleIgnore}}$:

```
$\underline{\mathit{Sample}(R \subset \{0,1\}^\lambda)}$:
1. $r \leftarrow \{0,1\}^\lambda \setminus R$
2. Output $r$
```

We will now show that this is true, *if the attacker has to write down every element of R before giving it to Sample.*

1. Devise a strategy how you would distinguish both libraries. Write down an adversary, as an algorithm, to do this. You can use one or more queries to *Sample*. Let's call this algorithm $\mathcal{A}$.

2. Assume your algorithm $\mathcal{A}$ makes 1 call to *Sample*, where $|R| = n$. For which outputs of $L_{\textsc{GenSample}}$ will $\mathcal{A}$ output 1 with the same probability as when $\mathcal{A}$ talks to $L_{\textsc{GenSampleIgnore}}$? And for which outputs of $L_{\textsc{GenSample}}$ will $\mathcal{A}$ output something different? How does this translate into the advantage of $\mathcal{A}$ making one query?

3. Assume that $\mathcal{A}$ makes 2 calls to *Sample*. What is maximal distinguishing advantage for $\mathcal{A}$ now?

4. Generalize your argument to $q \in \mathbb{N}$ calls to *Sample*. You can use the trick from the Birthday Bound in the lecture to give an upper bound on the success probability.

5. Assume that your attacker $\mathcal{A}$ makes $q \in \mathbb{N}$ calls to *Sample*, and that generating each item of R for each call takes 1 unit of time. Argue that since $\mathcal{A}$ has a runtime that is polynomial in λ, then this translates into a polynomial number of calls q and a polynomial size of R in each call (using the restriction on R outlined above). Then, conclude why the distinguishing advantage must be negligible.