Problem sheet 2 for Course 012231, 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 $1/|K|$.

For the first exercise, we denote with & the AND-function, which is defined by the following truth table:

x	y	$x\&y$
0	0	0
0	1	0
1	0	0
1	1	1

For two vectors $\mathbf{x}, \mathbf{y}$ we can define the coordinate-wise AND of both vectors, $\mathbf{x}\&\mathbf{y}$, by saying that $(\mathbf{x}\&\mathbf{y})[i] = \mathbf{x}[i]\&\mathbf{y}[i]$. This means that $\mathbf{x}\&\mathbf{y}$ is computed by applying & to the individual coordinates of each $\mathbf{x}, \mathbf{y}$ and concatenating the outcomes.

❓ Exercise 1. **(Not all bit operations are equal)**

Let λ be a positive integer. We define $K = M = C = \{0,1\}^\lambda$ and consider the following algorithms:

```
KEYGEN():
    1. Output $\mathbf{k} \leftarrow K$.
ENC($\mathbf{k}, \mathbf{m} \in M$):
    1. Output $\mathbf{k}\&\mathbf{m}$.
DEC($\mathbf{k}, \mathbf{c} \in C$):
    1. Output $\mathbf{k}\&\mathbf{c}$.
```

Show that these three algorithms do *not* form a SKE scheme, because they are not correct. You can show this by giving an example where decryption fails.

For the second exercise, we will work with permutations on bitstrings. A permutation of a string is a rearrangement of all the entries of that string. For example, let $a, b \in \{0,1\}$ be bits and ab be a bitstring. Then both ab and ba are permutations of the original string. Note that aa is not a permutation of ab, as b also has to appear in the permutation of ab. In general, there are $n!$ permutations of n objects. For an n-bit string, there are thus $n!$ permutations (but they are not all distinct).

Mapping all bitstrings ab to ab is a function from $\{0,1\}^2$ to $\{0,1\}^2$, and similarly is mapping all bitstrings ab to ba. For all strings of length n, we denote the set of permutations (or rather, all different functions that shuffle the n entries of input vectors) as S_n.

For a string $\mathbf{x} \in \{0,1\}^n$ we write $\mathbf{y} := \pi(\mathbf{x})$ to say that $\mathbf{y}$ is the string $\mathbf{x}$ permuted under the permutation function $\pi \in S_n$. Formally, this means that $\mathbf{y}[\pi(i)] = \mathbf{x}[i]$ for all $i \in \{1, \dots, n\}$. Accordingly, let π^{-1} be the inverse permutation. Note that this inverse permutation always exists: if we shuffle the indices of a vector in a certain way, we can just unshuffle them by going backwards.

❓ Exercise 2. (Permutations for Security?)

Let λ be a positive integer and let S_λ be the set of all permutations on strings of length λ. We define $M = C = \{0,1\}^\lambda$ and consider the following algorithms:

$\textsc{Keygen}()$:

1. Output $\pi \leftarrow S_\lambda$.

$\textsc{Enc}(\pi, \mathbf{m} \in M)$:

1. Output $\pi(\mathbf{m})$.

$\textsc{Dec}(\pi, \mathbf{c} \in C)$:

1. Output $\pi^{-1}(\mathbf{c})$.

1. Show that these three algorithms form an SKE scheme, i.e. that they are correct.
2. Let L_{OTS-0}, L_{OTS-1} be the two worlds in the left-or-right one-time security game described in the lecture and the book. Show that the defined three algorithms are not one-time indistinguishable, i.e. that $L_{OTS-0} \not\equiv L_{OTS-1}$. For this, write down an explicit algorithm B which sends one query to the library that it has access to black box. Based on the response c, B outputs 0 if it thinks that it talked to L_{OTS-0} or 1 if it thinks that it talked to L_{OTS-1}. How likely is it that B gives the right answer?

For the last exercise, we denote with $\oplus$ the XOR-function, which is defined by the following truth table:

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

As for the AND-function, we can define it coordinate-wise for vectors.

Exercise 3. (**Why Keygen is important**)

Let λ be a positive integer. We define $K = M = C = \{0,1\}^\lambda$ and let $\mathcal{B}_p$ be the probability distribution that outputs 0 with probability p and 1 with probability $1-p$. Consider the following algorithms:

$\textsc{Keygen}()$**:**

1. Let $\mathbf{k}$ be a vector of length λ.
2. For each $i \in \{1, \dots, \lambda\}$ set $\mathbf{k}[i] \leftarrow \mathcal{B}_{0.75}$

$\textsc{Enc}(\mathbf{k}, \mathbf{m} \in M)$**:**

1. Output $\mathbf{k} \oplus \mathbf{m}$.

$\textsc{Dec}(\mathbf{k}, \mathbf{c} \in C)$**:**

1. Output $\mathbf{k} \oplus \mathbf{c}$.

1. Show that these three algorithms form an SKE scheme, i.e. that they are correct.
2. Let $\lambda = 3$ and let A be an algorithm which queries the $L_{OTS-Real}$ library with input $\mathbf{m} = 111$. Compute the probability of each possible ciphertext obtained from the library.
3. Based on the algorithm A, consider a new algorithm B that sends $\mathbf{m} = 111$, waits for the response $\mathbf{c}$ from the library. B outputs 1 if $\mathbf{c} = \mathbf{m}$ and 0 otherwise. What is the probability that B outputs 1 with $L_{OTS-Real}$, and what is the probability when it interacts with $L_{OTS-Rand}$?
4. Show that the three algorithms $\textsc{Keygen}, \textsc{Enc}, \textsc{Dec}$ are not real-or-random secure, i.e. that $L_{OTS-Real} \not\equiv L_{OTS-Rand}$. For this, apply the real-or-random-secure definition to the algorithm B and show why it breaks security.
5. * **Bonus problem:** Is B the attacker with the highest probability of guessing correctly? If not, find an optimal attack.