Homework 1 for 01410, 2023 (10 points)

Exercise 1.1 (5 points) When using a block cipher to encrypt arbitrary-length messages, it is necessary to *pad* the message so that its length is a multiple of the block length $n < 8 \cdot 2^8$, which we assume to be a multiple of 8. A possible padding scheme for messages consisting of an integer number of bytes is as follows.

PKCS #5 padding algorithm.

- Input: a message $m \in \{0,1\}^*$ of length $8 \cdot l$, where l is a positive integer.
- Output: a message $m' \in \{0,1\}^*$ such that the length of m' is an integer multiple of n.
- Let $\hat{l} > 0$ be the smallest positive integer such that $8(l + \hat{l})$ is a multiple of n. Set $m' = m \| \underbrace{[\hat{l}] \| [\hat{l}] \| \ldots \| [\hat{l}]}_{\hat{l} \text{ times}}$, where $[\hat{l}]$ denotes the byte that represents the integer $\hat{l}$.

When decrypting a ciphertext, a check is performed if the padding is correct. If it is not, an error message is returned. Otherwise, the padding is removed and the message is returned.

1. (2 points) The above padding algorithm always appends at least one byte, and sometimes $n/8$ bytes. Show that this is necessary, i.e. prove that for any padding algorithm $\mathcal{P}$ where at most $n/8 - 1$ bytes are appended, there are two messages m_1 and m_2 that are mapped to the same message m' by $\mathcal{P}$. For simplicity, you can consider messages of at most $n/8$ bytes, i.e. messages that fit into one block.
2. (3 points) Suppose Alice and Bob use AES in CBC mode with PKCS #5 padding to send messages back and forth. Whenever Alice receives a ciphertext with bad padding, she asks Bob to re-send the message, and vice versa. Suppose you are a "person in the middle",[1] i.e. you can modify any ciphertext coming from one of the two before it reaches the other. Describe an attack that allows you to learn the length of a message that, say, Alice sends to Bob. **Hint:** Take a close look at how the CBC-mode decryption of the last ciphertext block is done. Can you modify the ciphertext in a way such that the last block of the decrypted message and the last block of the original message differ only on one byte position?

Exercise 1.2 (3 points)

We have discussed in the lecture that S-boxes are important for block ciphers because they make the encryption function non-linear. Let $S : \{0,1\}^s \to \{0,1\}^s$ be a function that we would like to use as an S-Box. The S-box should be invertible.

One precise condition that it needs to fulfill is that it should not be affine. S is said to be affine if for all $x, y, a \in \{0,1\}^s$ we have that

$$S(x) \oplus S(y) = S(x \oplus a) \oplus S(y \oplus a).$$

1. (1 point) Prove that all invertible functions from 1 bit to 1 bit are affine

[1] Such an attacker is more commonly known as a "man in the middle", but I see no reason why the gender of the person should play a role.

2. (2 points) Give an example of a 3 bit to 3 bit invertible function that is not affine.

Remark: All functions from 2 bits to 2 bits are affine as well, thus an S-box needs to act on at least 3 bits.

Exercise 1.3 (2 points)
For one-time pad encryption of an n-bit message m, an n-bit key k is chosen uniformly at random, $k \leftarrow \{0,1\}^n$. As the key is uniformly random, it can happen that $k = 0^n = \underbrace{00\ldots0}_{n \text{ times}}$, in which case the message $c = m$ is transmitted! In an attempt to make the scheme more secure, consider the variant of the one-time pad where a non-zero k is picked uniformly at random, i.e. $k \leftarrow \mathbb{K} = (\{0,1\}^n \setminus \{0^n\})$.

Show that this encryption scheme does not fulfill perfect secrecy. To that end, exhibit a message $m \in \{0,1\}^n$ and a ciphertext $c \in \{0,1\}^n$ such that

$$\Pr_{k \leftarrow \mathbb{K}}[c = m \oplus k] = 0.$$

Explain how this implies that the modified one-time pad does not fulfill perfect secrecy.

What you should do

- Enrol in one of the homework submission groups. You are encouraged to work in groups of (up to) 3, so the groups have a capacity of 3.
- Write the solutions to the exercises in one document.
- Upload your document on Learn.
- You may work in groups of at most three students.
- The format of your document should be PDF.
- If you use program code of any kind, please include it **and** describe your solution to that it can be understood without looking at the code.