Secure and Dependable Systems
Jacobs University Bremen
Dr. Florian Rabe

Homework 6
given: 2017-04-27

Homework 6

You have to submit your solutions as announced in the lecture.
Unless mentioned otherwise, all problems are due 2017-05-04, before the lecture.
There will be no deadline extensions unless mentioned otherwise in the lecture.

This homework is not published yet. I may still change it before publishing it.

Problem 6.1 *Practice: Building an Encryption Scheme* Points: 5

Implement abstract classes for

- symmetric encryption schemes
- block ciphers

Implement concrete classes for

- the block cipher from the example in the lecture
- the encryption scheme that takes a block cipher and the IV and uses the CBC mode of operation (Every instance of the scheme should represent one session, i.e., subsequent calls of encrypt for the same block should return different results).

Write a unit test that checks the inversion condition: randomly generates some blocks, encrypt and decrypt them, and check for equality.

Problem 6.2 *Practice: Relevance of Modes of Operation* Points: 2

Use your implementation from the previous problem to encrypt a file.

This should be a real file in an uncompressed format, e.g., a bitmap image. It should be big enough to consist of many blocks.

Modify your implementation to use the trivial mode of operation (where no IV is used and each block is simply passed to the block cipher). Encrypt the same file with this mode and compare both results with the original.

Note: This homeworks aims at reproducing the effect from the penguin image example at https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Electronic_Codebook_.28ECB.29.

Problem 6.3 *Theory: Analysing the security of a block cipher* Points: 3

Suppose we are given a pseudo-random generator, that given $n \in \mathbb{N}$ produces a pseudo-random key of size n. No we construct the following encryption algorithm E_k. We assume that the message m to encrypt consists of m blocks of 16-bit each, otherwise we append 0s to the message to fill up the last block. We ask the pseudo-random generator for a sequence k_i of m sub-keys of 16-bit each.

Now we encrypt the m_i with the following block cipher E_{k_i} using the i-th sub-key:

- Define the substitution $S : \mathbb{Z}_{16} \to \mathbb{Z}_{16}, x \longmapsto (x+1) \cdot 7 (\text{mod} 17) - 1$. It can be shown (you are allowed to simply assume that this is true) using standard arguments of number theory, that this is in fact a bijection.
- Define the permutation $P : \mathbb{Z}_{16} \to \mathbb{Z}_{16}, y \longmapsto P(y)$ as a simple cyclic left-shift of the binary representation of y by 1 bit.
- Define the bijection addKey : $\mathbb{Z}_{16} \to \mathbb{Z}_{16}, z \longmapsto z \otimes k_i (\text{mod} 16)$.
- Finally define $E_{k_i} : \mathbb{Z}_{16} \to \mathbb{Z}_{16}, m_i \longmapsto E_{k_i}(m_i) := \text{addKey} \circ P \circ S(m_i)$.

The block cipher E_k is now simply the block cipher E_{k_i} run on each individual block (ecb-mode).

Solve either of the following two problems about the security of E_k:

1. Argue informally, why E_k is computationally indistinguishable.

Hint:

It might help to first show that addKey applied to each individual block is already computationally indistinguishable.

2. (Challenging) Explain why E_k is **not** IND_CPA secure.

Hint:

E_k can be broken deterministically using a chosen-plaintext attack. Since the security of E_k is mainly based on addKey and thus directly on the randomness of the sub-key, it might be useful to think about recovering the key using a chosen-plaintext attack.