Exercise Sheet 10

Meet-in-the-Middle

Obligatory homework: Exercise 5. Hand-in by November 28, 23:59.

Exercise 1: Double DES - Part 1
DES is only using a key size of 56 bits which makes it vulnerable to bruteforce attacks. It was suggested to use two 56 bits keys and double encryption to increase the keyspace and increase the resistance against those attacks, called *2DES*. The encryption is done in the following way

$$c = E_{k_2}(E_{k_1}(m))$$

- What is the complexity for an exhaustive key search?
- Can you apply a meet-in-the-middle attack to reduce the complexity?
- Would the system be more secure when using $c = E_{k_2}(E_{k_1}(E_{k_1}(m)))$.
- Would the system be more secure when using $c = E_{k_1}(E_{k_2}(E_{k_1}(m)))$.

Exercise 2: Double DES - Part 1
DES has a blocksize of 64 bits. When using 2DES with two 56-bit keys:

- What is the expected number of key pairs k_1, k_2 such that $E_{k_1}(m_1) = D_{k_2}(c_1)$?
- What is the expected number of key pairs k_1, k_2 such that $E_{k_1}(m_1) = D_{k_2}(c_1)$ and $E_{k_1}(m_2) = D_{k_2}(c_2)$?

Exercise 3: Triple DES
Another mode, 3DES, which is still used in practice, uses three 56-bit keys and encrypts a message in the following way:

$$c = E_{k_3}(D_{k_2}(E_{k_1}(m)))$$

- What is the complexity for an exhaustive key search?
- Can you still apply a meet-in-the-middle attack?

Exercise 4: MITM on a Simple Feistel Cipher
In this excercise you should implement a meet-in-the-middle attack on a simple block cipher. The block cipher operates on 16-bit blocks and is based on the Feistel structure (see Figure 1). It uses 8-bit independent round keys k_i and S is the 8-bit AES S-Box. See the C code (`mitm.c`) on CampusNet.

- Given the following plaintext/ciphertext pairs, implement a meet-in-the-middle attack on this cipher and recover the key:

Rounds	Plaintext	Ciphertext
4	`0000`	`4748`
	`1234`	`3cf6`

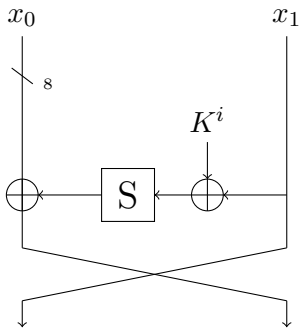

Figure 1: One round of the feistel cipher used in this exercise, where S is the AES S-Box.

Exercise 5: MITM on double DES
In this excercise your task is to implement a meet-in-the-middle attack on double-DES (see Figure 2). For this exercise you will use a reduced size for the keys. The two keys k_1 and k_2 are 20-bit keys padded with 0 to get a 56-bit key.

- Find an implementation of DES in the programming language of your choice.
- Choose two random 20-bit keys k_1, k_2 and pad them with zeroes to get a key for double-DES.
- Encrypt a plaintext of your choice using double-DES and your key (consisting of two chunks 20 bits each).

Your task is now to recover the secret keys k_1 and k_2 from your plaintext/ciphertext pair using a meet-in-the-middle attack.

- Implement a meet-in-middle attack on this scheme to retrieve k_1 and k_2.
- How long does it take to recover the key?
- What is the effective key length of this scheme?
- Give the memory/time complexity for the attack.

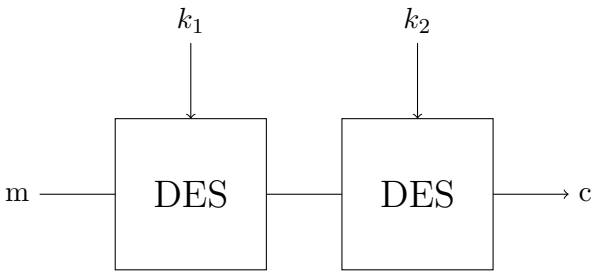

Figure 2: Double DES

Your hand-in (via CampusNet) should include:

1. source code of your two attack implementations (exercise 5),
2. a short report containing your results and discussion of these (remember to answer the questions posed in the exercises).