Introduction to Cryptography - Exercise session 5

Prof. Sebastian Faust

November 21, 2018

In the first part of this exercise, we recall the new topics covered during the lecture: modes of operation ECB, CBC and CTR, and the blockcipher DES. The second part of this sheet contains more interesting exercises.

PART 1

Exercise 1 (Modes of operation)

Recall the three modes of operation discussed during the lecture, i.e. ECB mode, CBC mode and CTR mode.

(a) Let F be a blockcipher with n-bit key and block length. For each of the modes write down/draw how a message $m_1, \ldots, m_\ell \in \{0,1\}^{\ell \times n}$ would be encrypted using F. For each mode, explain how decryption work.

Solution:

ECB mode

The ECB mode is an encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:

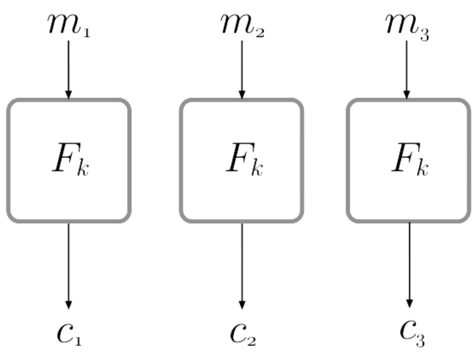
$\mathsf{Gen}(1^n)$**:** outputs a key $k \leftarrow_\$ \{0,1\}^n$

$\mathsf{Enc}_k(m_1, \ldots, m_\ell)$**:**

1. For $i = 1, .., \ell$ compute $c_i := F_k(m_i)$
2. Output $(c_1, \ldots, c_\ell)$.

$\mathsf{Dec}_k(c_1, \ldots, c_\ell)$**:**

1. For $i = 1, .., \ell$ compute $m_i := F_k^{-1}(c_i)$.
2. Output $(m_1, \ldots, m_\ell)$.



CBC mode

The CBC mode is an encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:

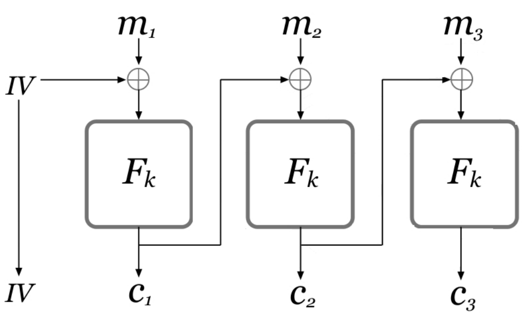
$\mathsf{Gen}(1^n)$**:** outputs a key $k \leftarrow_\$ \{0,1\}^n$

$\mathsf{Enc}_k(m_1, \ldots, m_\ell)$**:**

1. Sample uniformly at random $IV \leftarrow_\$ \{0,1\}^n$ and set $c_0 := IV$
2. For $i = 1, .., \ell$ compute $c_i := F_k(c_{i-1} \oplus m_i)$
3. Output $(c_0, c_1, \ldots, c_\ell)$.

$\mathsf{Dec}_k(c_0, c_1, \ldots, c_\ell)$**:**

1. For $i = 1, .., \ell$ compute $m_i := F_k^{-1}(c_i) \oplus c_{i-1}$.
2. Output $(m_1, \ldots, m_\ell)$.



CTR mode

The CTR mode as defined above is an encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:

$\mathsf{Gen}(1^n)$**:** outputs a key $k \leftarrow_\$ \{0,1\}^n$

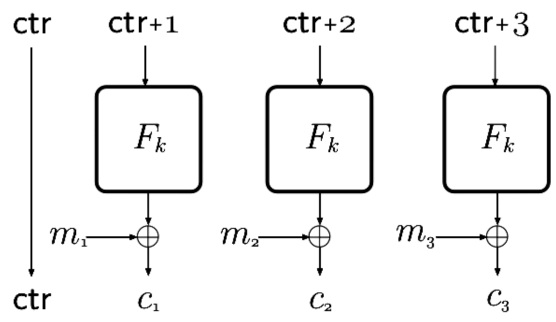
$\mathsf{Enc}_k(m_1, \ldots, m_\ell)$**:**

1. Sample uniformly at random $\mathsf{ctr} \leftarrow_\$ \{0,1\}^n$ and set $c_0 := \mathsf{ctr}$.
2. For $i = 1, .., \ell$ compute $c_i := m_i \oplus F_k(\mathsf{ctr} + i \text{ (modulo } 2^n))$
3. Output $(c_0, c_1, \ldots, c_\ell)$.

$\mathsf{Dec}_k(c_0, c_1, \ldots, c_\ell)$**:**

1. For $i = 1, .., \ell$ compute $m_i := c_i \oplus F_k(\mathsf{ctr} + i \text{ (modulo } 2^n))$.

2. Output $(m_1, \ldots, m_\ell)$.



(b) For each of the modes, explain the effect of a single-bit error in the ciphertext.

Solution:
Let us consider the following cipher text blocks $(c_0, c_1, \ldots, c_\ell) := \mathsf{Enc}(m_1, \ldots, m_\ell)$. At the receiver's end, cipher text is received as $(c'_0, c'_1, \ldots, c'_\ell)$, where

- $c'_j \neq c_j$ and $c'_j \oplus c_j = 2^x$ for some $x \in \{0, \ldots, l-1\}$
- $c'_i = c_i$ for $i \in \{0, 1, \ldots, \ell\} \setminus \{j\}$

ECB mode

By definition,

$$(m'_0, \ldots, m'_\ell) := \mathsf{Dec}(c'_0, c'_1, \ldots, c'_\ell) = (F_k^{-1}(c'_0), \ldots, F_k^{-1}(c'_\ell)).$$

Hence, we have that $m_i = m'_i \Leftrightarrow c_i = c'_i$. To conclude, exactly one block will be decrypted to a wrong message, i.e. $m'_j \neq m_j$ and for every $i \in \{0, \ldots, \ell\} \setminus \{j\}$, $m'_i = m_i$.

CBC mode

Let $\{m'_1, \ldots, m'_\ell\}$ be the result of the decryption. By definition, we know that for every $i \in \{0, 1, \ldots, \ell\}$

$$m'_i = F_k^{-1}(c'_i) \oplus c'_{i-1}.$$

Hence $m'_i = m_i \Leftrightarrow c'_i \oplus c'_{i-1} = c_i \oplus c_{i-1}$. This implies the following:

1. For $i \notin \{j, j+1\}$, $m'_i = m_i$.
2. $m'_j = F_k^{-1}(c'_j) \oplus c'_{j-1} = F_k^{-1}(c'_j) \oplus c_{j-1} \neq F_k^{-1}(c_j) \oplus c_{j-1} = m_j$
3. $m'_{j+1} = F_k^{-1}(c'_{j+1}) \oplus c'_j = F_k^{-1}(c_{j+1}) \oplus c'_j \neq F_k^{-1}(c_{j+1}) \oplus c_j = m_{j+1}$

In conclusion, two blocks will be decrypted wrongly, i.e. $\{m'_j,\, m'_{j+1}\}$. For every $i \in \{0, \ldots, \ell\} \setminus \{j, j+1\}$, $m'_i = m_i$.

CTR mode

By definition, we know that for every $i \in \{0, 1, \ldots, \ell\}$

$$m'_i = c'_i \oplus F_k(\mathsf{ctr} + i \text{ (modulo } 2^n))$$

Hence we have:

1. For $i \neq j$ it holds that $m_i = c_i \oplus F_k(\mathsf{ctr} + i \text{ (modulo } 2^n)) = c'_i \oplus F_k(\mathsf{ctr} + i \text{ (modulo } 2^n)) = m'_i$.
2. $m'_j = c'_j \oplus F_k(ctr + j) = (c_j \oplus 2^x) \oplus F_k(ctr + j) = m_j \oplus 2^x \neq m_j$

In conclusion, only one block is decrypted to a wrong message with a single bit error, i.e. $m'_j = m_j \oplus 2^x \neq m_j$ and for every $i \in \{0, \dots, \ell\} \setminus \{j\}$, $m'_i = m_i$.

Exercise 2 (DES)

Let F be a block cipher with n-bit key and ℓ-bit block length. Then the new block cipher F' with key of length $2n$ can be defined as

$$F'_{k_1,k_2}(x) := F_{k_2}(F_{k_1}(x)),$$

where k_1, k_2 are independent keys. For the case when $F = \mathsf{DES}$, we call $F' = 2\mathsf{DES}$. The above construction can be generalized to triple encryption as follows:

$$F''_{k_1,k_2,k_3}(x) := F_{k_3}(F^{-1}_{k_2}(F_{k_1}(x))).$$

If $F = \mathsf{DES}$, then the blockcipher F'' is called $3\mathsf{DES}$. The reason why the second invocation of F is reversed is for backward compatibility.

(a) Show how to design DES from $3\mathsf{DES}$.

Solution:
Let $k_1 = k_2 = k_3 = k$. The we have

$$3\mathsf{DES}_{k,k,k}(x) = \mathsf{DES}_k(\mathsf{DES}^{-1}_k(\mathsf{DES}_k(x))) = \mathsf{DES}_k(x).$$

(b) Show how to design $2\mathsf{DES}$ from $3\mathsf{DES}$.

Solution:
$3\mathsf{DES}_{k_2,k_2,k_2}(3\mathsf{DES}_{k_1,k_1,k_1}(x)) \overset{\text{Part } (a)}{=} \mathsf{DES}_{k_2}(\mathsf{DES}_{k_1}(x)) = 2\mathsf{DES}_{k_1,k_2}(x)$.

(c) Assume that F is a strong PRP. Informally argue, why the above construction of F'' is as good as if the second invocation of F would not be reversed, i.e. $F_{k_3}(F_{k_2}(F_{k_1}(x)))$.

Solution:
Recall the the adversary trying to distinguish a strong PRP from a random computation has random oracle access to both $F_k(\cdot)$ and $F^{-1}_k(\cdot)$ before the challenge phase. This immediately implies that if F is a strong PRP, then also F^{-1} is a strong PRP.

Exercise 3 (CBC mode)

Consider a stateful variant of the CBC-mode encryption Π where the sender simply increments the $IV \in \{0,1\}^n$ by 1 each time a message is encrypted (rather than choosing IV at random each time). Show that the resulting scheme is not CPA-secure.

Solution:

We design an adversary $\mathcal{A}$ that wins the CPA experiment with probability greater than $1/2 + 1/p(n)$, where p is some positive polynomial. The adversary $\mathcal{A}$ is defined as follows:

1. Query the encryption oracle with $m = 0^{n-1}1$ (binary string with $n-1$ zeros and 1 one) and receive cipher text $(IV, c) \in \{0,1\}^{2n}$.
2. If IV is odd, i.e. has as last bit 1, then output a random bit
3. If IV is even, i.e. has as last bit 0, then output $m_0 = 0^n$ (a bitstring consisting of n zeros) and arbitrary message $0^n \neq m_1 \in \{0,1\}^n$ to be encrypted.
4. Receive the challenge ciphertext $(IV + 1, c') \in \{0,1\}^{2n}$, and output 0 if $c' = c$, and 1 otherwise.

In order to bound $\mathcal{A}$'s success probability, let us distinguish two cases: the case when IV is odd and the case when IV is even. By the law of total probability, it holds that

$$\begin{aligned}\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] = &\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1 | IV \text{ odd}] \Pr[IV \text{ odd}] + &(1)\\ &\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1 | IV \text{ even}] \Pr[IV \text{ even}] &(2)\end{aligned}$$

In case IV is odd the adversary $\mathcal{A}$ outputs a random bit; hence his success probability is $\frac{1}{2}$, i.e.

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1 | IV \text{ odd}] = 1/2 \tag{3}$$

In case IV is even, we know that IV' is equal to $IV \oplus 0^{n-1}1$. In other words, the first $n-1$ bits of IV' are the same as the first $n-1$ bits of IV and the last bit of IV' is equal to 1. Therefore,

$$c' = F_k(IV' \oplus m_0) = F_k(IV \oplus 0^{n-1}1 \oplus 0^n) = F_k(IV \oplus 0^{n-1}1) = F_k(IV \oplus m) = c.$$

This implies that $c = c'$ if and only if m_0 was encrypted. Hence, $\mathcal{A}$ always decided correctly in case IV is even, i.e.

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1 | IV \text{ even}] = 1. \tag{4}$$

Using Eq. (3) and (4) and the fact that $\Pr[IV \text{ odd}] = \Pr[IV \text{ even}] = 1/2$, we can continue the calculation (1) as

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] = 1/2 \cdot 1/2 + 1 \cdot 1/2 = 1/2 + 1/4.$$

Exercise 4 (Meet-in-the-middle attack)

Let F be a block cipher with n-bit key and ℓ-bit block length. Consider a block cipher F' with key of length $2n$ defined as

$$F'_{k_1,k_2}(x) := F_{k_2}(F_{k_1}(x)),$$

where k_1, k_2 are independent n-bit keys.

(a) Design an adversary that given only one valid (plaintext, ciphertext) pair (x, y), i.e.

$$y = F'_{k_1^*,k_2^*}(x),$$

can find a set S consisting of all key pairs (k_1, k_2) such that $y = F'_{k_1,k_2}(x)$ and whose time complexity is asymptotically smaller that the time complexity of the bruteforce attack (which is $\mathcal{O}(2^{2n})$). Hint: Make use of the name of this exercise.

> **Solution:**
> The adversary tries to recover the secret key (k_1^*, k_2^*) as follows:
>
> 1. For each $k_1 \in \{0,1\}^n$, compute $z_1 := F_{k_1}(x)$ and store (z_1, k_1) in a list L_1. Time complexity of this step is $\mathcal{O}(2^n)$.
> 2. For each $k_2 \in \{0,1\}^n$, compute $z_2 := F_{k_2}^{-1}(y)$ and store (z_2, k_2) in a list L_2. The time complexity of this step is $\mathcal{O}(2^n)$.
> 3. Entries (z_1, k_1) and (z_2, k_2) are called a *match* if $z_1 = z_2$. For each such match, add (k_1, k_2) to a set S. Note that for every $(k_1, k_2) \in S$, the following is true
>
> $$F_{k_1}(x) = F_{k_2}^{-1}(y) \iff y = F'_{k_1,k_2}(x).$$
>
>   Finding the *match* pairs is easy if both list are sorted. Since the time complexity of sorting alg. is $\mathcal{O}(\log N \cdot N)$, where $N = 2^n$ is the size of the list, the time complexity of this step is $\mathcal{O}(n \cdot 2^n)$
>
> The overall time complexity is therefore $\mathcal{O}(n \cdot 2^n)$.

(b) What is the space complexity of the above algorithm?

> **Solution:**
> The adversary has to store the two lists L_1, L_2. Each of them consists of 2^n pairs, where the first entry has bit length n and the second one ℓ. Hence, the overall space complexity is equal to $\mathcal{O}((n + \ell) \cdot 2^n)$.

(c) Assume that the adversary knows two pliantext, ciphertext pairs (x_1, y_1) and (x_2, y_2) for $x_1 \neq x_2$, i.e. $y_1 = F'_{k_1^*,k_2^*}(x_1)$ and $y_2 = F'_{k_1^*,k_2^*}(x_2)$. Does this additional knowledge help the attacker? Explain your answer.

> **Solution:**
> Yes, the amount of candidate keys decreases.
>
> The attacker can run the above algorithm twice, once with (x_1, y_1) and once with (x_2, y_2). As a result, he obtains two sets S_1 and S_2. Since both pairs were generated

using the same key k_1^*, k_2^*, it must hold that $(k_1^*, k_2^*) \in S_1 \cap S_2$ and since $x_1 \neq x_2$, then also $|S_1 \cap S_2| < \min\{|S_1|, |S_2|\}$.

HOMEWORK

Exercise 5 (Chained CBC)

Is the chained CBC mode scheme defined below CPA-secure? If not, illustrate with an attack.

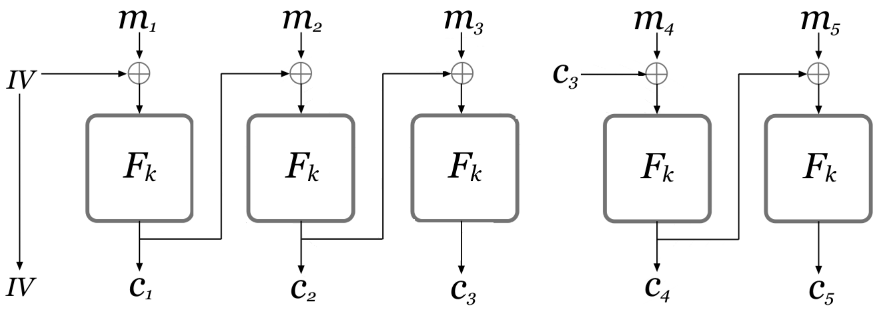


Figure 1: Chained CBC