

## SOLUTION OF EXERCISESHEET 6

### Exercise 6-1

(a)

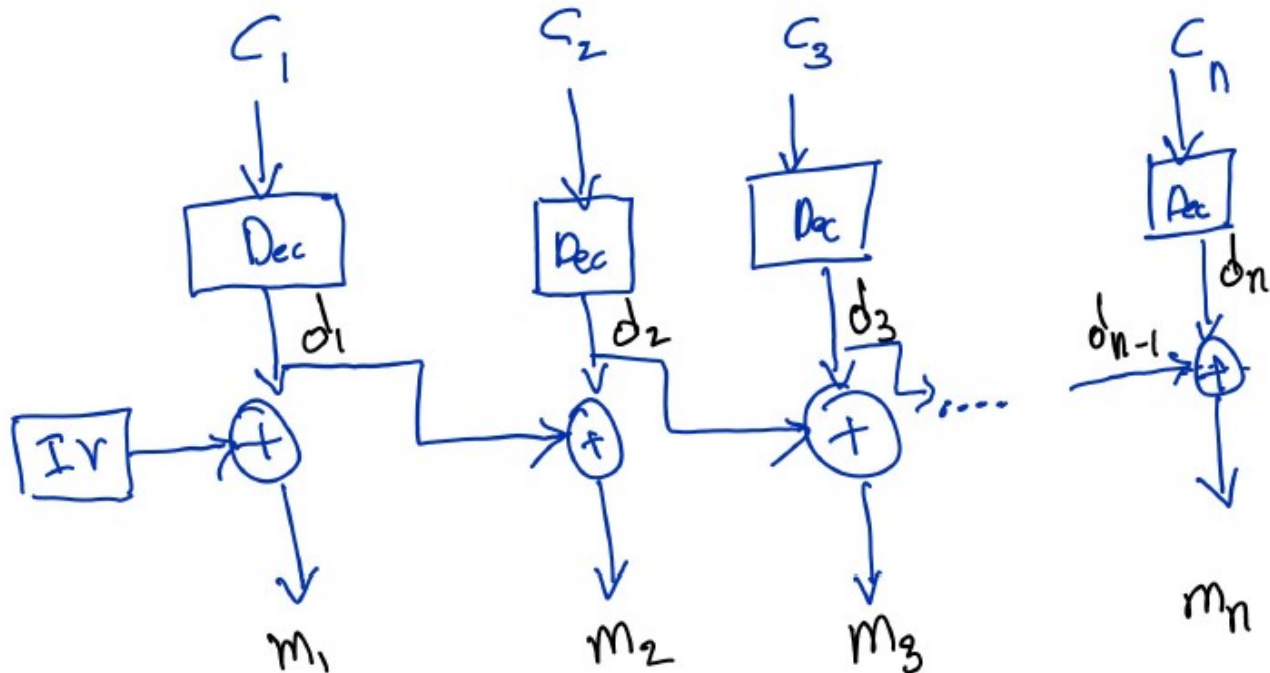


Figure 1: Decryption for CBC\* mode

(b) As shown in the above figure, let us assume

$$d_i = \text{Dec}(c_i)$$

So

$$\begin{aligned} m_1 &= d_1 \oplus IV \\ m_2 &= d_2 \oplus d_1 \dots \end{aligned}$$

To show that this CBC\* doesn't have indistinguishable encryptions, let us consider message in the format  $m = m_1 || m_2 || m_3 || \dots || m_n$ . Also we know for CPA, adversary  $A$  is allowed of multiple encryptions.

Let us consider  $A$  choose two messages i.e.,  $m_1$  and  $m_2$

$$\begin{aligned} m_1 &= m_{1_1} || m_{1_2} || m_{1_3} || \dots || m_{1_n} \\ m_2 &= m_{2_1} || m_{2_2} || m_{2_3} || \dots || m_{2_n} \end{aligned}$$



## SOLUTION OF EXERCISESHEET 6

And  $m_1$  is chosen in such a way that  $m_{1_1} = m_{1_2} = m_{1_3} = \dots = m_{1_n}$  and  $m_2$  is chosen in such a way that  $m_{2_1} \neq m_{2_2} \neq m_{2_3} \neq \dots \neq m_{2_n}$

With these kind of messages chosen,  $A$  can distinguish  $m_1$  and  $m_2$  by checking

$$\begin{aligned} c_1 &= c_3 = \dots = c_i \\ c_2 &= c_4 = \dots = c_{i+1} \\ \text{where } i &\text{ is an odd number } \leq n \end{aligned}$$

If the above check is satisfied then the cipher  $c$  corresponds to  $m_1$ . Else it corresponds to  $m_2$ . With this construction  $A$  can distinguish between the messages with a probability equal to 1.

### Exercise 6-2

**Task:** Show that  $\Pi_{CBC}$  is not CCA-secure by demonstrating a successful adversary.

Assume  $n = 3$

The adversary  $\mathcal{A}$  can choose the two messages  $m_0 = m_0^1 || m_0^2 = 000\ 000$  and  $m_1 = m_1^1 || m_1^2 = 111\ 111$  which he sends to the challenger. Then he gets the ciphertext  $c_b = (c_b^0 || c_b^1 || c_b^2) = (IV || f_k(IV \oplus m_b^1) || f_k(f_k(IV \oplus m_b^1) \oplus m_b^2))$  back.

Then  $\mathcal{A}$  flips the last bit from  $c_b^2$ , so  $(c_b^2)' = c_b^2 \oplus 001$  and asks the decryption oracle for the decryption of  $c'_b = c_b^0 || c_b^1 || (c_b^2)'$ . Because  $c'_b \neq c_b$  the decryption oracle answers with  $m' = f_k^{-1}(c_b^1) \oplus c_b^0 || f_k^{-1}(c_b^2) \oplus c_b^1 = f_k^{-1}(f_k(IV \oplus m_b^1)) \oplus IV || f_k^{-1}((c_b^2)') \oplus f_k(IV \oplus m_b^1) = m_b^1 || f_k^{-1}((c_b^2)') \oplus f_k(IV \oplus m_b^1)$ .  $m_b^1$  is now either  $m_0^1$  or  $m_1^1$  because the change in  $(c_b^2)'$  doesn't impact  $m_b^1$ . So the adversary can say for sure, if the received ciphertext  $c_b$  is the encoding for  $m_0$  or  $m_1$ .

$\Rightarrow \Pi_{CBC}$  mode is not CCA-secure

### Exercise 6-3

## SOLUTION OF EXERCISESHEET 6

a)

Let  $F$  be a pseudorandom permutation. Then  $F$  and  $F^{-1}$  are pseudorandom permutations.

$\Pi_M = (\text{Gen}, \text{Mac}, \text{Vrfy})$

$\frac{\text{Gen}(1^\lambda)}{k \leftarrow \text{Gen}(1^\lambda)}$   
**return**  $k$

$\frac{\text{Mac}_k(c)}{t \leftarrow F_k^{-1}(c)}$   
**return**  $t$

$\frac{\text{Vrfy}_k(c, t)}{\text{if } t = \text{Mac}_k(m)}$   
**return** 1  
**return** 0

$\Pi_E = (\text{Gen}, \text{Enc}, \text{Dec})$

$\frac{\text{Gen}(1^\lambda)}{k \leftarrow \text{Gen}(1^\lambda)}$   
**return**  $k$

$\frac{\text{Enc}_k(m)}{r \leftarrow \{0, 1\}^n}$   
 $c \leftarrow F_k(m \parallel r)$   
**return**  $c$

$\frac{\text{Dec}_k(c)}{v := F_k^{-1}(c)}$   
**return** first  $n$  bits of  $v$

### Proof that $\Pi_M$ is unforgeable

We reduce the security of the Mac to the pseudorandomness of the function  $F^{-1}$ .

Therefore, we first assume that the construction is not secure and therefore there exists an adversary  $\mathcal{A}$  that wins MacForge with non-negligible probability  $\varepsilon(\lambda)$ . We use this adversary  $\mathcal{A}$  to build a distinguisher for the pseudorandomness of  $F_k$ .

With the help of the oracle  $O_D$  of the pseudorandomness,  $\mathcal{D}$  answers the oracle requests of  $\mathcal{A}$  by computing  $t := O_D(m)$ . If the oracle answers with a pseudorandom function, the view of  $\mathcal{A}$  is identical to  $\text{MacForge}_{\mathcal{A}, \Pi}(\lambda)$ . Thus we have

$$\Pr [D^{F_k^{-1}(\cdot)}(1^\lambda) = 1] = \Pr [\text{MacForge}_{\mathcal{A}, \Pi}(\lambda) = 1] = \varepsilon$$

where  $k \leftarrow \{0, 1\}^\lambda$ .

If the oracle answers with a random function, then we simulate the game for a different MAC-scheme  $\Pi'$ . Let  $\Pi' = (\text{Gen}', \text{Mac}', \text{Vrfy}')$  be a message authentication code which is the same as  $\Pi$ , except it uses a truly random function  $f$  instead of the pseudorandom function  $F_k$ . It is easy to see that  $\Pr [\text{MacForge}_{\mathcal{A}, \Pi'}(\lambda) = 1] \leq 2^{-\lambda}$

This is the case because for any message  $m$ , the value  $t$  is uniformly distributed in  $\{0, 1\}^*$  from the point of view of  $\mathcal{A}$ . The view of  $\mathcal{A}$  is identical to  $\text{MacForge}_{\mathcal{A}, \Pi}(\lambda)$ . We have

$$\Pr [D^{f(\cdot)}(1^\lambda) = 1] = \Pr [\text{MacForge}_{\mathcal{A}, \Pi'}(\lambda) = 1] \leq \frac{1}{2^\lambda}$$

where  $f \leftarrow \text{Func}_\lambda$ .

The distinguisher can now distinguish between pseudorandom and truly random with non-negligible probability. As we assumed the function  $F^{-1}$  to be pseudorandom, this is a contradiction and thus such an adversary cannot exist. Hence the MAC construction is secure.

## SOLUTION OF EXERCISESHEET 6

### Proof that $\Pi_E$ is CPA-secure

We assume towards contradiction that the scheme  $\Pi_E$  is not CPA-secure.

If  $\Pi_E$  is not CPA-secure then there exists an adversary  $\mathcal{A}$  that succeeds in the CPA-game  $\frac{1}{2}$  with probability  $\frac{1}{2} + \varepsilon(\lambda)$  where  $\varepsilon$  is a non-negligible function.

We now use the ability of the adversary  $\mathcal{A}$  to create a distinguisher  $\mathcal{D}$  that can distinguish between the underlying pseudorandom function  $F$  and a randomly chosen function  $f$ .

The distinguisher  $\mathcal{D}$  gets as input  $\lambda$  and access to  $O_{\mathcal{D}}$  that runs either  $F$  or  $f$ .

$\mathcal{D}$  simulates an encryption oracle  $O_{Enc}$  to  $\mathcal{A}$ . It answers with  $Enc(k, m)$  on the input  $m$  where the function  $F$  is replaced with the oracle  $O_{\mathcal{D}}$ .

The encryption oracle either answers with  $c := F_k(m \parallel r)$  or  $c := f(m \parallel r)$ .

$\mathcal{A}$  then asks for the encryption of one of the two messages  $m_0$  and  $m_1$  with  $|m_0| = |m_1|$ .  $\mathcal{D}$  then samples a bit  $b \leftarrow \{0, 1\}$  and forwards  $c_b \leftarrow Enc_k(m_b)$  to  $\mathcal{A}$  where  $Enc_k(m_b)$  is realised like in the encryption oracle.  $\mathcal{D}$  then outputs  $b' = b$ .

$\mathcal{D}$  is efficient because it only forwards messages what can be done in constant time and invokes  $\mathcal{A}$  which is efficient.

To analyse the success we distinguish two cases:

If  $O_{\mathcal{D}}$  runs a pseudorandom permutation function  $f$  then  $\mathcal{D}$  perfectly simulates  $\Pi_E$  to  $\mathcal{A}$ .

$\Rightarrow Pr[\mathcal{D}^{f(\cdot)}(1^\lambda) = 1] = Pr[PrivK_{\Pi_E, \mathcal{A}}^{CPA} = 1] = \frac{1}{2} + non - negl(\lambda)$ , because  $\mathcal{A}$  is an efficient adversary against the CPA-security of  $\Pi_E$

If the oracle runs a randomly chosen function  $f^*$  and  $\mathcal{A}$  queries the encryption oracle at least  $q$  times we have  $Pr[\mathcal{D}^{f^*(\cdot)}(1^\lambda) = 1] = \frac{1}{2} + \frac{q(\lambda)}{2^\lambda}$ .

Now we subtract those two cases:

$$|Pr[\mathcal{D}^{f(\cdot)}(1^\lambda) = 1] - Pr[\mathcal{D}^{f^*(\cdot)}(1^\lambda) = 1]| = \left| \frac{1}{2} + non - negl(\lambda) - \frac{1}{2} - \frac{q(\lambda)}{2^\lambda} \right| = non - negl(\lambda) - \frac{q(\lambda)}{2^\lambda} = non - negl(\lambda).$$

So the distinguisher  $\mathcal{D}$  can distinguish between  $f$  and  $f^*$  with a non-negligible gap which is a contradiction to the pseudorandomness of  $f$ .

Therefore such an adversary  $\mathcal{A}$  against the CPA-security of  $\Pi_E$  cannot exist.

b)

### Proof that $\Pi'$ is not CCA-secure

Because  $Enc_k(m), Mac_k(Enc_k(m)) = F_k(m \parallel r), F_k^{-1}(F_k(m \parallel r)) = F_k(m \parallel r), (m \parallel r)$

When the adversary  $\mathcal{A}$  receives its challenge ciphertext  $c = (c', t)$ , it can easily recover the message and knows which of its two messages was encrypted.