Solution of Exercisesheet 6

Exercise 6-1

(a)

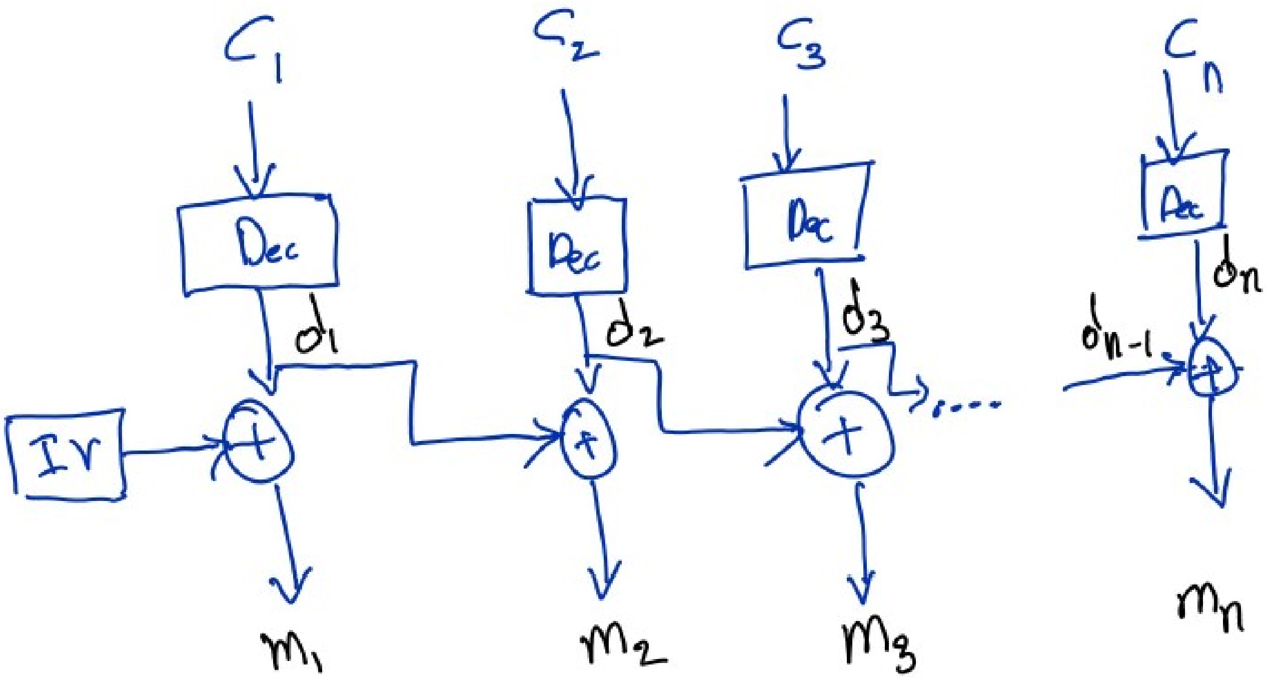

Figure 1: Decryption for CBC* mode

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

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

As given for CBC*,

$$d_1 = m_1 \oplus IV$$
$$d_2 = m_2 \oplus d_1$$
$$\text{i.e., } d_2 = m_2 \oplus m_1 \oplus IV$$
$$\text{Similarly, } d_3 = m_3 \oplus m_2 \oplus m_1 \oplus IV$$

The inverse for decryption would be,

$$m_1 = d_1 \oplus IV$$
$$m_2 = d_2 \oplus d_1 \text{ ...}$$

Solution of Exercisesheet 6

To show that this CBC* doesn't have indistinguishable encryptions, let us consider message in the format $m = m_1||m_2||m_3||...||m_n$.

Given the Enc is deterministic, the intution of the attack on this CBC* mode was that we can recognize repeated blocks in the message.

Let us consider IND-EAV adversary, A choose two messages i.e., $m1$ and $m2$ in the below format

$$m1 = m1_1||m1_2||m1_3||...||m1_n$$
$$m1 = m2_1||m2_2||m2_3||...||m2_n$$

And $m1$ is choosen in such a way that $m1_1 == m1_2 == m1_3 == ... == m1_n$ and $m2$ is choosen in such a way that $m2_1 \neq m2_2 \neq m2_3 \neq ... \neq m2_n$

Then

$$d1_1 = m1_1 \oplus IV$$
$$d1_2 = m1_2 \oplus d1_1$$
$$\text{i.e., } d1_2 = m1_1 \oplus m1_1 \oplus IV = IV$$
$$\text{Similarly, } d1_3 = m1_1 \oplus m_1 \oplus m1_1 \oplus IV = m1_1 \oplus IV$$

As Enc is deterministic, A can distingush $m1$ and $m2$ by checking Enc($m1$) and Enc($m2$) as

$$c_1 == c_3 == ... == c_i$$
$$c_2 == c_4 == ... == c_{i+1}$$
$$\text{where } i \text{ is an odd number} \leq n$$

If the above check is statisfied then the cipher c corresponds to $m1$. Else it corresponds to $m2$. With this construction A ouputs 1 if this check is statisfied else 0. This means A always wins CBC* as A runs in polynomial time.

Exercise 6-2

Task: Show that Π_{CBC} is not CCA-secure by demonstrating a successful adversary.

The adversary $\mathcal{A}$ can choose the two messages $m_0 = m_0^1||m_0^2 = 0^n\ 0^n$ and $m_1 = m_1^1||m_1^2 = 1^n\ 1^n$ 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 $(c_b^2)' = c_b^2 \oplus 0^{n-1}1$ 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 recieved civertext c_b is the encoding for m_0 or m_1.

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

Solution of Exercisesheet 6

Exercise 6-3

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

$\Pi_{\mathrm{M}} = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$

$\underline{\mathrm{Gen}(1^\lambda)}$	$\underline{\mathrm{Mac_k(c)}}$	$\underline{\mathrm{Vrfy_k(c,t)}}$
$k \leftarrow \mathrm{Gen}(1^\lambda)$	$t \leftarrow \mathrm{F}_k^{-1}(c)$	if $t = \mathrm{Mac}_k(m)$
return k	**return** t	**return** 1
		return 0

$\Pi_{\mathrm{E}} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$

$\underline{\mathrm{Gen}(1^\lambda)}$	$\underline{\mathrm{Enc_k(m)}}$	$\underline{\mathrm{Dec_k(c)}}$
$k \leftarrow \mathrm{Gen}(1^\lambda)$	$r \leftarrow \{0,1\}^n$	$v := \mathrm{F}_k^{-1}(c)$
return k	$c \leftarrow \mathrm{F}_k(m \parallel r)$	**return** first n bits of v
	return c	

Proof that Π_{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 $O_{\mathcal{D}}$ 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 $MacForge_{\mathcal{A},\Pi'}(\lambda)$. Thus we have

$$Pr\left[D^{F_k^{-1}(\cdot)}(1^\lambda) = 1\right] = Pr\left[MacForge_{\mathcal{A},\Pi}(\lambda) = 1\right] = \varepsilon$$

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

If the oracle answers with a random function, then we simulate the game for a different MAC-scheme Π'. Let $\Pi' = (Gen', Mac', Vrfy')$ be a message authentication code which is the same as Π, except it uses a truly random function f instead of the pseudorandom function F_k. It is easy to see that

$$Pr\left[MacForge_{A,\Pi'}(\lambda) = 1\right] \leq 2^{-\lambda}$$

Solution of Exercisesheet 6

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 $MacForge_{\mathcal{A},\Pi}(\lambda)$. We have
$Pr\left[D^{f(\cdot)}(1^\lambda)=1\right] = Pr\left[MacForge_{\mathcal{A},\Pi'}(\lambda)=1\right] \leq \frac{1}{2^\lambda}$
where $f \leftarrow Func_\lambda$.

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.

Proof that Π_E is CPA-secure
We assume towards contradiction that the scheme Π_E is not CPA-secure.

If Π_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 ε 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 λ 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 $c := \mathrm{F}_k(m \parallel r)$ 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 $\mathcal{O}_D$ runs a pseudorandom permutation function f then $\mathcal{D}$ perfectly simulates Π_E to $\mathcal{A}$.

$\Rightarrow Pr[\mathcal{D}^{f(\cdot)}(1^\lambda)=1] = Pr[PrivK^{CPA}_{\Pi^E,\mathcal{A}}=1] = \dfrac{1}{2} + non-negl(\lambda)$, because $\mathcal{A}$ is an efficient adversary against the CPA-security of Π_E

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

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

Solution of Exercisesheet 6

$negl(\lambda) - \frac{q(\lambda)}{2^\lambda} = non - negl(\lambda)$.
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 Π_E cannot exist.

b)

Proof that Π' is not CCA-secure

Because $\mathrm{Enc}_k(m), \mathrm{Mac}_k(\mathrm{Enc}_k(m)) = \mathrm{F}_k(m \parallel r), \mathrm{F}_k^{-1}(\mathrm{F}_k(m \parallel r)) = \mathrm{F}_k(m \parallel r), (m \parallel r)$

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.