SOLUTION OF EXERCISESHEET 7

Exercise 7-1

This modified encryption does not achieve CPA-security. $H(m)$ is only collision-resistant and doesn't have to hide the message m. If $H(m)$ leaks the message m and this is concatenated with $Enc(k,m)$ the resulting scheme $Enc'(k,m)$ can't be CPA-secure.

Exercise 7-2

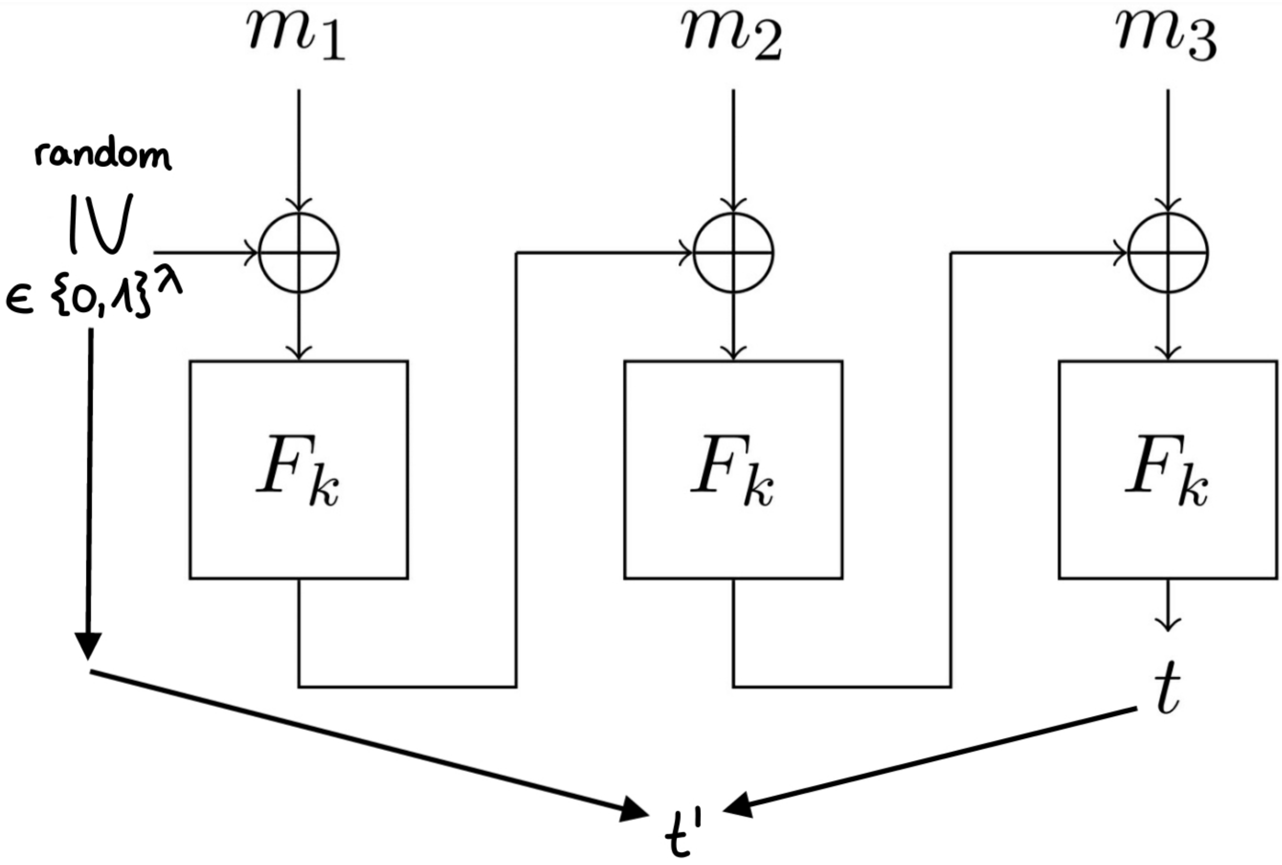
(a) not secure:
The adversary $\mathcal{A}$ makes two queries to the oracle:
$m^1 = m_1||m_2 \Rightarrow t^1 = t_1^1||t_2^1 = F(K, m_1)||F(K, F(K, m_2))$
$m^2 = F(K, m_1)||m_2 \Rightarrow t^2 = t_1^2||t_2^2 = F(K, F(K, m_1))||F(K, F(K, m_2))$
Then he knows the tag for the message $m^* = m_1||m_1$ which is $t^* = F(K, m1)||F(K, F(K, m1)) = t_1^1||t_1^2$. Because $m^* \neq m^1$ and $m^* \neq m^2$, (m^*, t^*) is a valid attack.

(b) not secure:
The adversary $\mathcal{A}$ makes one query to the oracle:
$m^1 = m_1||m_2 \Rightarrow t^1 = F(K, m_1) \oplus F(K, m_2)$
Then he knows the tag for the message $m^* = m_2||m_1$ which is $t^* = F(K, m_2) \oplus F(K, m_1) = F(K, m_1) \oplus F(K, m_2) = t^1$. Because $m^* \neq m^1$, (m^*, t^*) is a valid attack.

(c) not secure:
The adversary $\mathcal{A}$ makes one query to the oracle:
$m^1 = m_1||m_2 \Rightarrow t^1 = (r \oplus (F(K, m_1) \oplus F(K, m_2)), r)$
Then he knows the tag for the message $m^* = m_2||m_1$ which is $t^* = (r \oplus (F(K, m_2) \oplus F(K, m_1)), r) = (r \oplus (F(K, m_1) \oplus F(K, m_2)), r) = t^1$. Because $m^* \neq m^1$, (m^*, t^*) is a valid attack.

SOLUTION OF EXERCISESHEET 7

Exercise 7-3



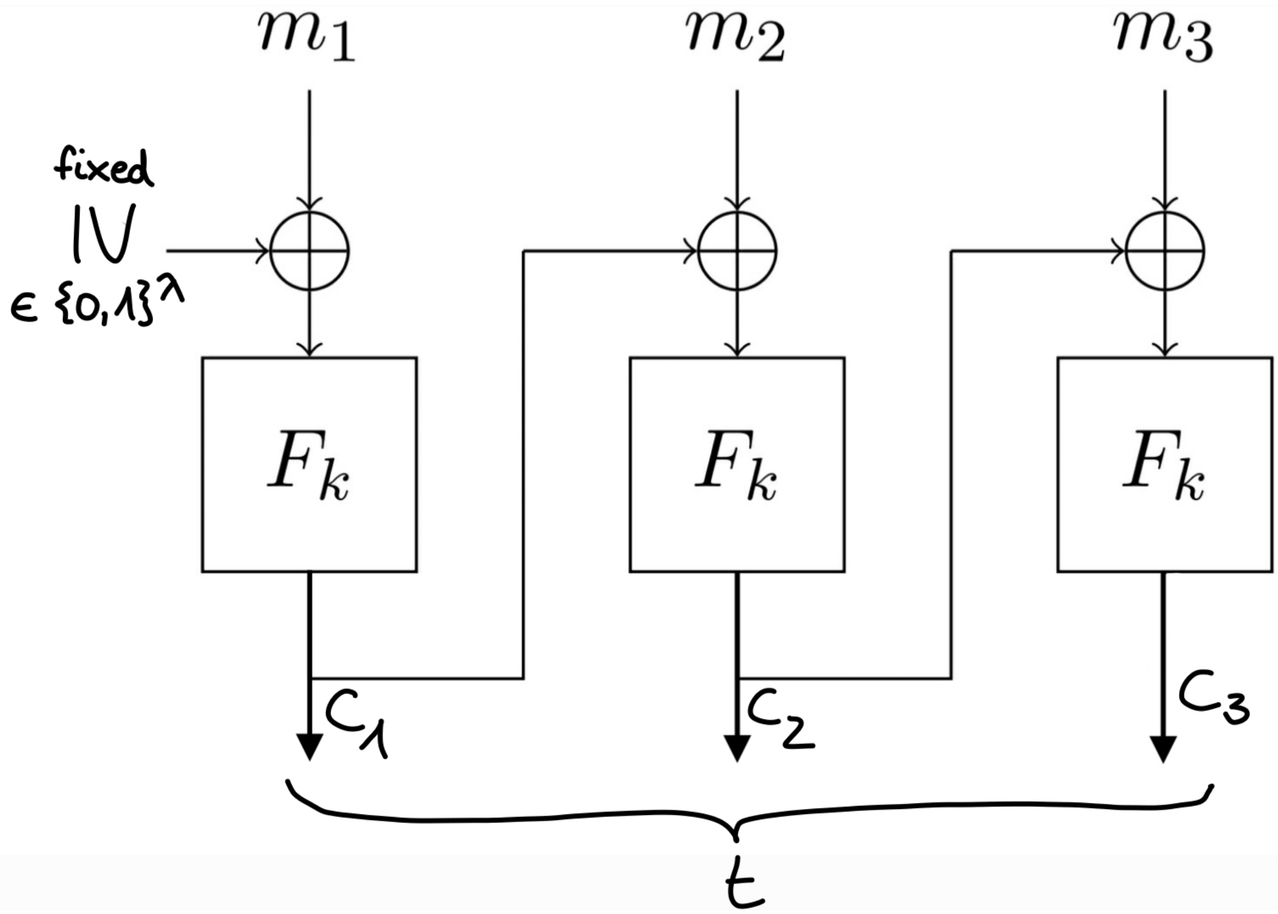
(a) The adversary $\mathcal{A}$ can choose the message m^* of length $\ell(\lambda) \cdot \lambda$. We set $\ell(\lambda) = l$
$m^* = m_1^* \,||\, ... \,||\, m_l^*$. We set $m_i^* = 0^\lambda$.
We query m^* to our $Mac_k(\cdot)$ oracle and recieve the tag $t'^* = (IV^*, t^*)$. $IV^* \in \{0,1\}^\lambda$ is a randomly generated vector.

$c_1^* = F_k(IV^* \oplus m_1^*) = F_k(IV^* \oplus 0^\lambda) = F_k(IV^*)$
$c_2^* = F_k(c_1^* \oplus m_2^*) = F_k(F_k(IV^*) \oplus 0^\lambda) = F_k(\ F_k(IV^*))$
...
$c_l^* = F_k(\ ...\ F_k(IV^*)\ ...) = t^*$

To break the security of the MAC the adversary $\mathcal{A}$ can construct the following forgery:
$m = 0^{\lambda-1}1 \,||\, m_2^* \,||\, ... \,||\, m_l^*$ with $IV = IV^* \oplus 0^{\lambda-1}1$
meaning we flip the last bit of the first block of our message as well as the last bit of our IV^*.
This causes that $IV^* \oplus m_1^* = IV \oplus m_1$ and therefore
$c_1^* = F_k(IV^* \oplus m_1^*) = F_k(IV \oplus m_1) = c_1$ as well as all other $c_i^* = c_i$ of the chain for $i \in \{2,l\}$.

We can conclude that $t^* = t$ but $m^* \neq m$ and therefore m was not queried to Mac_k before.
$\Rightarrow$ The adversary $\mathcal{A}$ can now create the valid forgery (m, t') with $t' = (IV, t^*)$ and can break the unforgability of the Mac.

Solution of Exercisesheet 7



(b) The adversary $\mathcal{A}$ can choose the messages m of length $\ell(\lambda) \cdot \lambda$. We set $\ell(\lambda) = l$. We choose
$m^1 = m^1_1 \mid\mid \dots \mid\mid m^1_l$ and
$m^2 = m^2_1 \mid\mid \dots \mid\mid m^2_l$
with $m^1_j = m^2_j = 0^\lambda$ for $2 \leq j \leq l$.

$c^i_1 = F_k(IV \oplus m^i_1)$ with IV $^* \in \{0,1\}^\lambda$ as a fixed vector
$c^i_2 = F_k(c^i_1 \oplus m^i_2)$
...
$c^i_l = F_k(\dots F_k(IV \oplus m^i_1) \dots)$
$t = (c^i_1, \dots, c^i_l)$ for $i \in \{1,2\}$

To break the security of the MAC the adversary $\mathcal{A}$ can construct the following forgery:
$m = m^2_1 \mid\mid m_2{}' \mid\mid m^1_3 \mid\mid \dots \mid\mid m^1_l$
with $m_2{}' = c^1_1 \oplus m^1_2 \oplus c^2_1$ which generates a vector that satisfies the following equation:
$\Rightarrow c^1_1 \oplus m^1_2 = c^2_1 \oplus m_2{}'$
This helps us construct m such that $c^1_2 = F_k(c^1_1 \oplus m^1_2) = F_k(c^2_1 \oplus m_2{}') = c_2$ and $c^1_i = c_i$ for $i \in \{3,l\}$.
Therefore the adversary $\mathcal{A}$ can create the valid forgery (m, t) with $t = (c^2_1, c^1_2, \dots, c^1_l)$ for m which has not been queried before and can break the unforgability of the Mac.

Solution of Exercisesheet 7

Exercise 7-4

Π' is unforgeable under an adaptive chosen message attack.
For proof, let's assume Π' is not unforgeable under an adaptive chosen message attack. Then there exists and adversary $\mathcal{A}$ who is able to find a valid message, tag pair in polynomial time.
We use $\mathcal{A}$ to build an adversary $\mathcal{D}$ who breaks Π_M, an unforgeable MAC.
$\mathcal{D}$ runs $\mathcal{A}$. When $\mathcal{A}$ seconds a message to Mac'_k, the message is hashed with $H^s(m)$ and input to MAC. MAC returns a tag t which is returned to $\mathcal{A}$.
$\mathcal{A}$ eventually outputs a valid message, tag pair in polynomial time. $\mathcal{D}$ therefore also outputs a valid message, tag pair in polynomial time. $\mathcal{D}$ is efficient because $\mathcal{A}$ is efficient and messages are only forwarded and hashed what happens in polynomial time too.