TD Bonus (corrected version)

Exercise 1. *CTR Security*

Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a PRF. To encrypt a message $M \in \{0,1\}^{d \cdot n}$, CTR proceeds as follows:

- Write $M = M_0 \| M_1 \| \dots \| M_{d-1}$ with each $M_i \in \{0,1\}^n$.
- Sample IV uniformly in $\{0,1\}^n$.
- Return $IV \| C_0 \| C_1 \| \dots \| C_{d-1}$ with $C_i = M_i \oplus F(k, IV + i \bmod 2^n)$ for all i.

The goal of this exercise is to prove the security of the CTR encryption mode against chosen plaintext attacks, when the PRF F is secure.

1. Recall the definition of security of an encryption scheme against chosen plaintext attacks.

☞ Let (KeyGen, Enc, Dec) be an encryption scheme. We consider the following experiments Exp_b for $b \in \{0,1\}$:

- Challenger samples $k \leftarrow$ KeyGen,
- Adversary makes q encryption queries on messages $(M_{i,0}, M_{i,1})$,
- Challenger sends back $Enc(k, M_{i,b})$ for each i,
- Adversary returns $b' \in \{0,1\}$.

We define the advantage of the adversary $\mathcal{A}$ against the encryption scheme as

$$\mathsf{Adv}^{\mathsf{CPA}}(\mathcal{A}) = \left| \Pr(\mathcal{A} \xrightarrow{\mathsf{Exp}_1} 1) - \Pr(\mathcal{A} \xrightarrow{\mathsf{Exp}_0} 1) \right|.$$

Then, the encryption scheme is said to be secure against chosen plaintext attacks if no probabilistic polynomial-time adversary has a non-negligible advantage with respect to n.

(Note in particular that since $\mathcal{A}$ runs in polynomial time, q must be polynomial in n.)

Remark: in another equivalent definition, there is only one experiment in which the challenger starts by choosing the bit b uniformly at random, and the advantage is defined as $\mathsf{Adv}^{\mathsf{CPA}}(\mathcal{A}) = |\Pr(\mathcal{A} \to 1 \mid b = 0) - \Pr(\mathcal{A} \to 1 \mid b = 1)|$.

2. Assume an attacker makes Q encryption queries. Let $IV_1, \dots, IV_Q$ be the corresponding IV's. Let $\texttt{Twice}$ denote the event "there exist $i, j \le Q$ and $k_i, k_j < d$ such that $IV_i + k_i = IV_j + k_j \bmod 2^n$ and $i \neq j$." Show that the probability of $\texttt{Twice}$ is bounded from above by $Q^2 d / 2^{n-1}$.

☞ *Remark: the probability of* $\texttt{Twice}$ *is obviously* 1 *if it is not required that* i *and* j *be distinct. Besides, considering the case* $i = j$ *is not interesting for our purpose.*

For $i, j \le Q$, let $\texttt{Twice}_{i,j}$ be the event "$\exists k_i, k_j < d : \mathrm{IV}_i + k_i = \mathrm{IV}_j + k_j \pmod{2^n}$", which is equivalent to "$\exists k, |k| < d$ and $\mathrm{IV}_i - \mathrm{IV}_j = k \pmod{2^n}$". As the IVs are chosen uniformly and independently, $\mathrm{IV}_i - \mathrm{IV}_j$ is uniform modulo 2^n and $\Pr(\texttt{Twice}_{i,j}) \le 2^{-n}(2d-1)$. (The inequality is strict when $2d - 1 > 2^n$, in which case $\Pr(\texttt{Twice}_{i,j}) = 1$.) Then,

$$\Pr(\texttt{Twice}) \le \sum_{1 \le i \neq j \le Q} \Pr(\texttt{Twice}_{i,j}) = Q(Q-1)2^{-n}(2d-1) \le 2^{1-n} Q^2 d.$$

3. Assume the PRF F is replaced by a uniformly chosen function $f : \{0,1\}^n \to \{0,1\}^n$. Give an upper bound on the distinguishing advantage of an adversary $\mathcal{A}$ against this idealized version of CTR, as a function of d, n and the number of encryption queries Q.

☞ We write $M^{i,\beta} = M_0^{i,\beta} \| \dots \| M_{d-1}^{i,\beta}$ with $1 \le i \le Q$ and $\beta \in \{0,1\}$ the encryption queries of the adversary $\mathcal{A}$ and $C^i = \mathrm{IV}_i \| C_0^i \| \dots \| C_{d-1}^i$ with $1 \le i \le Q$ the replies. Given the value of $b \in \{0,1\}$ chosen by the challenger, we know that $C_j^i = M_j^{i,b} \oplus f(\mathrm{IV}_i + j \pmod{2^n})$ for all $1 \le i \le Q$ and $0 \le j < d$.

If $\texttt{Twice}$ does not occur, then all the $\mathrm{IV}_i + j \pmod{2^n}$ for $1 \le i \le Q$ and $0 \le j < d$ are pairwise distinct. Then the values of f at these points are independent and uniformly distributed, since $f : \{0,1\}^n \to \{0,1\}^n$ is chosen uniformly at random. Therefore, all the C_j^i are also independent and uniformly distributed regardless of the value of b, so that $\Pr(\neg\texttt{Twice} \wedge \mathcal{A} \to 1 \mid b = 0) = \Pr(\neg\texttt{Twice} \wedge \mathcal{A} \to 1 \mid b = 1)$. It follows that

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{U}}^{\mathsf{CPA}}(\mathcal{A}) &= |\Pr(\texttt{Twice} \wedge \mathcal{A} \to 1 \mid b = 0) - \Pr(\texttt{Twice} \wedge \mathcal{A} \to 1 \mid b = 1)| \\
&= |\Pr(\mathcal{A} \to 1 \mid b = 0, \texttt{Twice}) - \Pr(\mathcal{A} \to 1 \mid b = 1, \texttt{Twice})| \Pr(\texttt{Twice}) \\
&\le \Pr(\texttt{Twice}) \le 2^{1-n} Q^2 d.
\end{aligned}$$

4. Show that if there exists a probabilistic polynomial-time adversary $\mathcal{A}$ against CTR based on PRF F, then there exists a probabilistic polynomial-time adversary $\mathcal{B}$ against the PRF F. Give a lower bound on the advantage degradation of the reduction.

☞ Assume that $\mathcal{A}$ is a PPT adversary against the encryption scheme with a non-negligible advantage for a chosen plaintext attack. We build an adversary $\mathcal{B}$ against the underlying PRF F as follows:

1. Choose $b \in \{0,1\}$ uniformly at random.
2. For each encryption query (M^0, M^1) from $\mathcal{A}$, encrypt M^b using the given scheme, that is,
   (a) Choose $\mathrm{IV} \in \{0,1\}^n$ uniformly at random.
   (b) For $j = 0$ to $d-1$, send a query for $\mathrm{IV} + j$ and with the reply f_j compute $C_j = M_j^b \oplus f_j$.
   (c) Send $\mathrm{IV} \| C_0 \| \ldots \| C_{d-1}$ back to $\mathcal{A}$.
3. When $\mathcal{A}$ finally outputs a bit $b' \in \{0,1\}$, output 1 if $b' = b$ and 0 otherwise.

The advantage of $\mathcal{B}$ against the PRF F is

$$\mathsf{Adv}_F^{\mathsf{PRF}}(\mathcal{B}) = |\Pr(\mathcal{B} \to 1 \mid \mathsf{PRF}) - \Pr(\mathcal{B} \to 1 \mid \mathsf{Unif})|$$

where PRF is the experiment in which replies to $\mathcal{B}$ are computed by calling F and Unif is the one in which replies to $\mathcal{B}$ are computed from a uniformly chosen random function f.

Considering the two terms separately gives

$$\begin{aligned}\Pr(\mathcal{B} \to 1 \mid E) &= \frac{1}{2}\left(\Pr(b' = 0 \mid E, b = 0) + \Pr(b' = 1 \mid E, b = 1)\right) \\ &= \frac{1}{2}\left(1 + \Pr(\mathcal{A} \to 1 \mid E, b = 1) - \Pr(\mathcal{A} \to 0 \mid E, b = 0)\right)\end{aligned}$$

where E is either PRF or Unif. Therefore

$$\mathsf{Adv}_F^{\mathsf{PRF}}(\mathcal{B}) \geq \frac{1}{2}\left(\mathsf{Adv}^{\mathsf{CPA}}(\mathcal{A}) - \mathsf{Adv}_{\mathcal{U}}^{\mathsf{CPA}}(\mathcal{A})\right) \geq \frac{1}{2}\mathsf{Adv}^{\mathsf{CPA}}(\mathcal{A}) - 2^{1-n}Q^2 d$$

using the previous question. Thus, if $\mathsf{Adv}^{\mathsf{CPA}}(\mathcal{A})$ is non-negligible then so is $\mathsf{Adv}_F^{\mathsf{PRF}}(\mathcal{B})$, which is then about a half of $\mathsf{Adv}^{\mathsf{CPA}}(\mathcal{A})$.

Exercise 2. *weak PRF*

In the PRF security game, the adversary may adaptively make function evaluation queries: for $i = 1, 2, \ldots$, it sends x_i of its choice, and gets $F_k(x_i)$ (resp. $f(x_i)$) from the challenger, where F_k is the PRF (resp. f is the uniformly chosen function). A weak-PRF consists of the same algorithms as a PRF, but the queries are modified as follows: the adversary does not get to see $F_k(x_i)$ (resp. $f(x_i)$) for **an input** x_i **of its choice**, but instead every time the adversary requests a new pair, **the challenger samples a fresh uniform** x_i and sends $(x_i, F_k(x_i))$ (resp. $(x_i, f(x_i))$) to the adversary.

1. Give a formal definition of a weak-PRF, based on a security game.

☞ A function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^d$ is a weak-PRF if for every efficient (e.g., ppt) adversary $\mathcal{A}$, we have that $Adv(\mathcal{A})^{wPRF} := |\Pr[\mathcal{A} \to 1 \text{ in } \mathsf{Exp}_{Real}] - \Pr[\mathcal{A} \to 1 \text{ in } \mathsf{Exp}_{Unif}]|$ is negligible. Exp_{Real} is when $\mathcal{C}$ samples k uniformly in $\{0,1\}^n$ and sets $f = F_k$ in the experiment below. Exp_{Real} is when $\mathcal{C}$ samples $f : \{0,1\}^m \to \{0,1\}^d$ uniformly.

Challenger $\mathcal{C}$		Algorithm $\mathcal{A}$
Samples f		
	$\longleftarrow$ ping	(as many times as desired)
Samples $x \leftarrow U(\{0,1\}^m)$		
$(x, f(x)) \longrightarrow$		
		Output a bit b.

2. Show that a PRF is a weak-PRF, by providing a security reduction.

☞ Here is the reduction:

PRF challenger $\mathcal{C}$	Algorithm $\mathcal{B}$		wPRF Attacker $\mathcal{A}$
Chooses f, either uniform or F_k			
		$\longleftarrow$ ping	(as many times as desired)
	Samples $x \leftarrow U(\{0,1\}^m)$		
	$\longleftarrow x$		
$(x, f(x)) \longrightarrow$			
	$\longrightarrow$		
			$\longleftarrow b$
	Output b		

When $\mathcal{C}$ uses F_k, the view of $\mathcal{A}$ is as in experiment Exp_{Unif} above. When $\mathcal{C}$ uses f, the view of $\mathcal{A}$ is as in experiment Exp_{wPRF} above. Hence $\mathsf{Adv}(\mathcal{B})^{PRF} = \mathsf{Adv}(\mathcal{A})^{wPRF}$.

3. Assuming that a weak-PRF exists, build a weak-PRF that is not a PRF.

☞ Let F be a secure weak-PRF. For all key k, we define F'_k as F_k, except that $F'_k(0^m) = 0^d$.

We have that F' is not a PRF, an adversary can query 0^m and output $b = 1$ if and only if the reply is 0^d. In the Real experiment, this adversary outputs $b = 1$ with probability 1. In the Unif experiment, it outputs $b = 1$ with probability $1/2^d$. The advantage is non-negligible.

Let us now argue that F' is still a weak PRF. The probability that during the experiment the challenger samples 0^m to answer one of the attacker's queries is $\leq Q \cdot 2^{-m}$, where Q is the number of queries made by the adversary. Let us call this event Bad. Assume we have an attacker $\mathcal{A}$ for F. We build an attacker $\mathcal{B}$ for F' as follows:

wPRF challenger $\mathcal{C}$ for F	Algorithm $\mathcal{B}$	wPRF Attacker $\mathcal{A}$ for F'
Chooses f, either uniform or F_k		
	$\longleftarrow$ ping	(as many times as desired)
	$\longleftarrow$ ping	
Sample $x \leftarrow U(\{0,1\}^m)$ $(x, f(x)) \longrightarrow$		
	$\longrightarrow$	
		$\longleftarrow b$
	Output b	

We have:

$$\begin{aligned}
\mathsf{Adv}(\mathcal{B} \text{ for } F) &= \Big| \Pr[\mathcal{B} \to 1 \text{ in } \mathsf{Exp}_{Unif} | Bad] \Pr[Bad] + \Pr[\mathcal{B} \to 1 \text{ in } \mathsf{Exp}_{Unif} | \overline{Bad}] \Pr[\overline{Bad}] \\
&\quad - \Pr[\mathcal{B} \to 1 \text{ in } \mathsf{Exp}_{Real} | Bad] \Pr[Bad] + \Pr[\mathcal{B} \to 1 \text{ in } \mathsf{Exp}_{Real} | \overline{Bad}] \Pr[\overline{Bad}] \Big| \\
&\leq \Pr[Bad] + \Pr[\overline{Bad}] \Big| \Pr[\mathcal{B} \to 1 \text{ in } \mathsf{Exp}_{Unif} | \overline{Bad}] - \Pr[\mathcal{B} \to 1 \text{ in } \mathsf{Exp}_{Real} | \overline{Bad}] \Big| \\
&= \Pr[Bad] + \Pr[\overline{Bad}] \Big| \Pr[\mathcal{A} \to 1 \text{ in } \mathsf{Exp}_{Unif} | \overline{Bad}] - \Pr[\mathcal{A} \to 1 \text{ in } \mathsf{Exp}_{Real} | \overline{Bad}] \Big| .
\end{aligned}$$

Note that the last term is $\leq \mathsf{Adv}(\mathcal{A} \text{ for } F')$. Hence:

$$\mathsf{Adv}(\mathcal{B} \text{ for } F) \leq Q \cdot 2^{-m} + \mathsf{Adv}(\mathcal{A} \text{ for } F').$$

4. What is the difference between a PRG and a weak-PRF?

☞ In a PRG experiment for a univariate function G, the challenger uniformly samples a (secret) seed s and sends $G(s)$ to the adversary. In a weak-PRF experiment for a bivariate function F, the challenger uniformly samples a (secret) key k, then for the Q queries of the attacker, is samples uniform x_i's and sends back to the attacker the xi's together with either $F(k, x_i)$. Note that if $Q = 1$, then the games are similar, and x_1 can even be considered as part of the description of G (formally, we can set $G(\cdot) = F(\cdot, x_1)$). So the main difference between a PRG and a weak-PRF is that in a weak-PRF the adversary can query as many inputs as it wants. This is different from the PRG case where the description of G is fixed and the size of the output if fixed (the adversary cannot ask for more).

Alternatively, one may compare $G(\cdot)$ and $F(k, \cdot)$: in the first case the seed s stays secret, in the second case the input x_i is provided to the adversary.

Let $G = (g)$ be a cyclic group of known prime order p. We recall that the DDH hardness assumption states that the distributions (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) are computationally indistinguishable when a, b and c are independently and uniformly distributed in $\mathbb{Z}/p\mathbb{Z}$. Let $k \in \mathbb{Z}/p\mathbb{Z}$ a uniformly chosen key. We consider the function $F_k : h \in G \mapsto h^k \in G$.

5. Let $Q \geq 1$. Consider the (randomized) map ϕ that takes $(g_1, g_2, g_3) \in G^3$ as input, samples $(x_i, y_i) \in (\mathbb{Z}/p\mathbb{Z})^2$ uniformly and independently for $i \leq Q$ and returns $(g_1^{x_i} g^{y_i}, g_3^{x_i} g_2^{y_i})_{i \leq Q}$.

   - Show that if $(g_1, g_2, g_3) = (g^a, g^b, g^{ab})$, then the output is distributed as $(g^{r_i}, g^{br_i})_{i \leq Q}$ for r_i's in $\mathbb{Z}/p\mathbb{Z}$ uniform and independent.
   - Show that if $(g_1, g_2, g_3) = (g^a, g^b, g^c)$ for $c \neq ab$, then the output is distributed as $(g^{r_i}, g^{s_i})_{i \leq Q}$ for (r_i, s_i)'s in $(\mathbb{Z}/p\mathbb{Z})^2$ uniform and independent.

☞ In the case where $c = ab$, we have
$$\left(g_1^{x_i} g^{y_i}, g_3^{x_i} g_2^{y_i}\right) = \left(g^{ax_i+y_i}, g^{abx_i+by_i}\right).$$
So, by letting $r_i = ax_i + y_i$, this is (g^{r_i}, g^{br_i}). Moreover, as the y_i's are uniform in $\mathbb{Z}_p$ and independent of the x_i's and a, the r_i's are also uniform. Finally, as the y_i's are all independent, then so are the r_i's.

In the case where $c \neq ab$, we have $(g_1^{x_i} g^{y_i}, g_3^{x_i} g_2^{y_i}) = (g^{r_i}, g^{s_i})$, where
$$\begin{pmatrix} r_i \\ s_i \end{pmatrix} = \begin{pmatrix} a & 1 \\ c & b \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}.$$
As $c \neq ab$ (and p is prime), the matrix is invertible. Hence, it induces a bijection over $\mathbb{Z}_p^2$. As the (x_i, y_i)'s are uniform and independent, we conclude that so are the (r_i, s_i)'s.

6. Show that F_k is a weak-PRF under the DDH hardness assumption.
Hint: set "$k = b$" and use the previous question to build the weak PRF challenger.

☞ Let $\mathcal{A}$ be a weak-PRF attacker against F. Let us build an algorithm $\mathcal{B}$ against the DDH assumption.

DDH challenger $\mathcal{C}$	Algorithm $\mathcal{B}$	wPRF Attacker $\mathcal{A}$
Sample a bit β, and $a, b, c \leftarrow U(\mathbb{Z}_p)$ If $\beta = 0$, then set $c = ab$ $(g^a, g^b, g^c) \longrightarrow$		
		$\longleftarrow$ ping (as many times as desired)
	$x_i, y_i \leftarrow U(\mathbb{Z}_p)$ $h_i = (g^a)^{x_i} \cdot g^{y_i}, t_i = (g^c)^{x_i} \cdot (g^b)^{y_i}$ store the values (h_i, t_i) and if some h_i shows up again, then replace t_i by the one that was obtained before. $(h_i, t_i) \longrightarrow$	
		$\longleftarrow \beta'$
	Output β'	

Let us analyze the above game. If $c = ab$, then for each query, algorithm $\mathcal{A}$ receives $h_i = g^{r_i}$ and $t_i = g^{br_i}$ where $b \leftarrow U(\mathbb{Z}_p)$ stays the same throughout the experiment. Moreover, as the r_i's are uniform in $\mathbb{Z}_p$ and independent, the h_i's are uniform and independent in G. So $\mathcal{A}$'s view is exactly the same as if it were given oracle access to F as in the weak-PRF game.

Now, if $c \neq ab$, adversary $\mathcal{A}$ receives $(h_i, t_i) = (g^{r_i}, g^{s_i})$, where the (r_i, s_i)'s are uniform and independent. So the (h_i, t_i)'s are also uniform and independent in G^2. Moreover the answers of $\mathcal{B}$ are consistent, meaning that each h_i always comes with the same t_i (that's why algorithm $\mathcal{B}$ is keeping a table!). Then the adversary's view is the same as if it were oracle access to a uniform map f.

To conclude, it holds that
$$\begin{aligned}
\mathrm{Adv}(\mathcal{B}) &= |\Pr(\beta' = 1|\beta = 1) - \Pr(\beta' = 1|\beta = 0)| \\
&= |\Pr(\beta' = 1|c = ab) - \Pr(\beta' = 1|c \leftarrow U(\mathbb{Z}_p))| \\
&= |\Pr(\beta' = 1|c = ab) - \Pr(\beta' = 1|c \neq ab)\Pr(c \neq ab|c \leftarrow U(\mathbb{Z}_p)) - \Pr(\beta' = 1|c = ab)\Pr(c = ab|c \leftarrow U(\mathbb{Z}_p))| \\
&= \frac{p-1}{p} \cdot |\Pr(\beta' = 1|c = ab) - \Pr(\beta' = 1|c \leftarrow U(\mathbb{Z}_p \setminus \{ab\}))| \\
&= \frac{p-1}{p} \cdot \mathrm{Adv}(\mathcal{A}).
\end{aligned}$$

Here, the last equality comes from the above discussion. Then if the DDH assumption holds, the advantage of $\mathcal{A}$ is negligible, and F is a secure weak-PRF.

7. Is F_k a secure PRF? Justify your answer.

☞ No. Consider the following adversary $\mathcal{A}$. It queries g and g^2 and gets two values x and x_2. It returns 1 if and only if $x_2 = x^2$ and 0 otherwise. In the PRF game, algorithm $\mathcal{A}$ always outputs 1. In the case of the uniform game, it is wrong if and only if $F(g^2) = F(g)^2$, which happens with probability $1/p$. Its advantage is then $\frac{p-1}{p}$, which is non-negligible.

Exercise 3. *CBC-MAC*

Let $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a PRF, $d > 0$ and $L = nd$. Prove that the following modifications of CBC-MAC (recalled in Figure 1) do not yield a secure fixed-length MAC. Define $t_i := F(K, t_{i-1} \oplus m_i)$ for $i \in [1, d]$ and $t_0 := IV = 0$.

1. Modify CBC-MAC so that a random $IV \hookleftarrow U(\{0,1\}^n)$ (rather than $IV = \mathbf{0}$) is used each time a tag is computed, and the output is (IV, t_d) instead of t_d alone.

☞ If an adversary asks for a tag (t_0, t_d) of any $(m_1, \ldots, m_d)$, then it can output $(t_0 \oplus x, t_d), (m_1 \oplus x, \ldots, m_d)$ as a forgery, as it is a valid pair of a tag and a message. Such an adversary wins everytime and has non-negligible advantage in the unforgeability game.

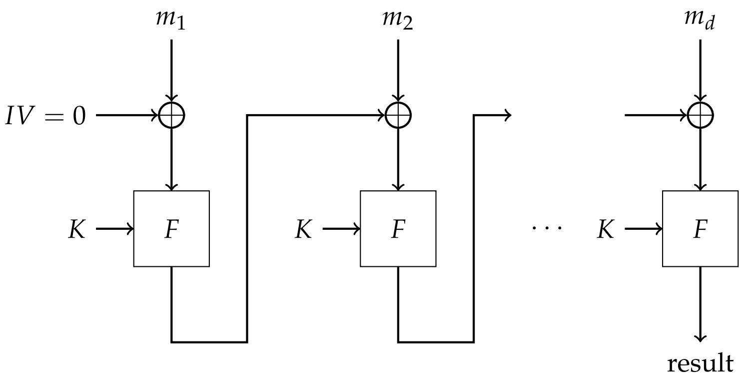

Figure 1: CBC-MAC

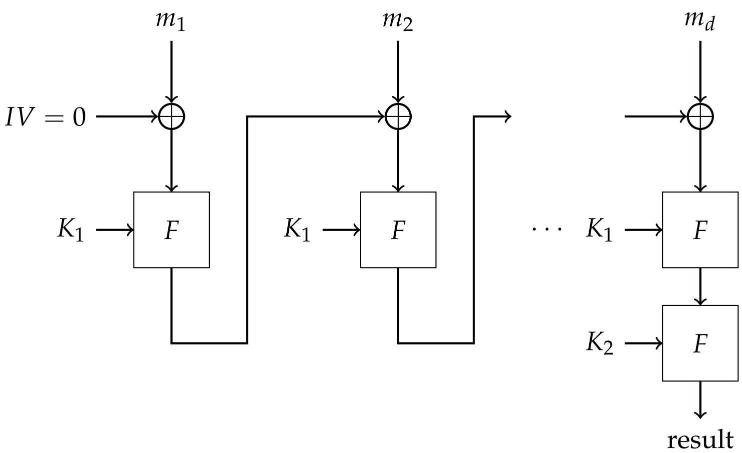

Figure 2: ECBC-MAC

2. Modify CBC-MAC so that all the outputs of F are output, rather than just the last one.
☞

If an adversary aks for a tag $(t_1, t_2, \ldots, t_d)$ of any message $(0, m_2, \ldots, m_d)$, then it can output $(t_2, t_3, \ldots, t_d, t_1), (m_2 \oplus t_1, m_3, \ldots, m_d, t_d)$ as a forgery as it is a valid pair (tag, message). Such an adversary wins everytime. Indeed, $F(K, m_2 \oplus t_1 \oplus 0) = t_2$ by definition and $F(K, t_d \oplus t_d) = t_1$ since $m_1 = 0$.

We now consider the following ECBC-MAC scheme: let $F : K \times X \to X$ be a PRF, we define $F_{ECBC} : K^2 \times X^{\leq L} \to X$ as in Figure 2, where K_1 and K_2 are two independent keys.
If the message length is not a multiple of the block length n, we add a pad to the last block: $m = m_1 | \ldots | m_{d-1} | (m_d \| \text{pad}(m))$.

3. Show that there exists a padding for which this scheme is not secure.
☞

We could for instance pad with as many 0s as necessary.
Let m of length $< n$. Then, $m||\text{pad}(m) = m||0||\text{pad}(m||0)$. As such we build an adverary for the unforgeability game that:

- asks for a tag for m of length $< n$.
- Gets a tag t.
- Returns the forgery $(m||0, t)$.

This adversary always wins and as such breaks the unforgeability of the scheme.

For the security of the scheme, the padding must be invertible, and in particular for any message $m_0 \neq m_1$ we need to have $m_0||\text{pad}(m_0) \neq m_1||\text{pad}(m_1)$. In practice, the ISO norm is to pad with $10 \cdots 0$, and if the message length is a multiple of the block length, to add a new "dummy" block $10 \cdots 0$ of length n.

4. Prove that this scheme is not secure if the padding does not add a new "dummy" block if the message length is a multiple of the block length.

☞ Let $m = m_1 \parallel 100$ of the length of a block, then $m = m_1 \parallel \mathrm{pad}(m_1)$, so any valid tag for m is a valid tag for m_1.

Remark: The NIST standard is called CMAC, it is a variant of CBC-MAC with three keys (k, k_1, k_2). If the message length is not a multiple of the block length, then we append the ISO padding to it and then we also XOR this last block with the key k_1. If the message length is a multiple of the block length, then we XOR this last block with the key k_2. After that, we perform a last encryption with $F(k, .)$ to obtain the tag.

Exercise 4. *Merkle-Damgård transform*

1. In the Merkle-Damgård transform, the message is split into consecutive blocks, and we add as a last block the binary representation of the length of this message. Suppose that we do not add this block: does this transform still lead to a collision-resistant hash function?

☞ No. Take for instance x of length $B\ell(n) - 1$ for some $B \geq 2$, and $y = x\|0$. In the transform, we start by padding x with one zero so that its length is a multiple of $\ell(n)$: we obtain y. In the rest of the process, the only thing that differs between x and y is that their "length blocks" are not the same; without this length block, x and y form a collision.

2. Before HMAC was invented, it was quite common to define a MAC by $\mathrm{Mac}_k(m) = H^s(k \parallel m)$ where H is a collision-resistant hash function. Show that this is not a secure MAC when H is constructed via the Merkle-Damgård transform.

☞ The goal is to construct (m, t) with $\mathrm{Verify}_k(m, t) = 1$, having oracle access to Mac_k but without querying $\mathrm{Mac}_k(m)$ itself.
With Merkle-Damgård, the function H^s divides the message $k \parallel m$ in p blocks $x_1, \ldots, x_p$ of size ℓ (padding the last block x_p with a Padding Block PB so that $x_p \parallel \mathrm{PB}$ has size ℓ) and then adding a new block x_{p+1} of length ℓ depending on the bit length of $k \parallel m$. Then the Merkle-Damgård construction uses a (fixed-length) collision-resistant hash function h to compute its output as follows:

$$H^s(k \parallel m) = h^s(x_{p+1}, h^s(x_p \parallel \mathrm{PB}, h^s(x_{p-1}, h^s(\ldots, h^s(x_1, \mathrm{IV}))))).$$

Given $H^s(k \parallel m)$, anyone can compute $H^s(k \parallel m \parallel \mathrm{PB} \parallel x_{p+1} \parallel \omega)$ for any ω; for instance, if ω is of size ℓ, using $h^s(x'_{p+2}, h^s(\omega, H^s(k \parallel m)))$ where x'_{p+2} only depends on the length of $k \parallel m \parallel PB \parallel x_{p+1} \parallel \omega$ and can be publicly computed.