Tutorial 5: CTR mode and MACs

Exercise 1. *Security of the CTR encryption scheme*

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 $M = M_0 \| M_1 \| \ldots \| M_{d-1}$ with each $M_i \in \{0,1\}^n$.
- Sample IV uniformly in $\{0,1\}^n$.
- Return $IV \| C_0 \| C_1 \| \ldots \| C_{d-1}$ 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 (many-time) chosen plaintext attacks, when the PRF F is secure.

1. Recall the definition of security of an encryption scheme against (many-time) chosen plaintext attacks.
2. Assume an attacker makes q encryption queries. Let $IV_1, \ldots, IV_q$ be the corresponding IV's. Let `Twice` denote the event "there exist $i \neq j \leq q$ and $k_i, k_j < d$ such that $IV_i + k_i = IV_j + k_j \bmod 2^n$." Show that the probability of `Twice` is bounded from above by $q^2 d / 2^n$.
3. Assume the PRF F is replaced by a uniformly chosen function $f : \{0,1\}^n \to \{0,1\}^n$. Bound the distinguishing advantage of an adversary $\mathcal{A}$ against this idealized version of CTR, as a function of d and the number of encryption queries q.
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.

Exercise 2. *MACs and PRFs*

1. We have seen that pseudo-random functions imply secure deterministic MACs for fixed-length messages.

   Give a construction of a secure deterministic MAC which is *not* a pseudo-random function.

2. Let F be a secure pseudorandom function (PRF). We consider the following message authentication code (MAC), for messages of length $2n$: The shared key is a key $k \in \{0,1\}^n$ of the PRF F; To authenticate a message $m_1 \| m_2$ with $m_1, m_2 \in \{0,1\}^n$, compute the tag $t = (F(k, m_1), F(k, (F(k, m_2))))$. Is it a secure MAC?

3. Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF. Consider the following MAC. To authenticate a message $m = m_1 \| m_2 \| \ldots \| m_d$ where $m_i \in \{0,1\}^n$ for all i, using a key k, compute

$$t = F(k, m_1) \oplus \ldots \oplus F(k, m_d).$$

   Is it a secure MAC?

Exercise 3. *MACs with verification oracle*

In the notion of existential **strong** unforgeability under chosen-message attacks, the adversary is given access to a MAC generation oracle $\mathsf{Mac}(K, .)$.

At each query M, the challenger computes $t \leftarrow \mathsf{Mac}(K, M)$, returns t and updates the set of MAC queries $Q := Q \cup \{(t, M)\}$, which is initialized to $Q := \emptyset$. At the end of the game, the adversary outputs a pair $(M^\star, t^\star)$ and wins if: (i) $\mathsf{Verify}(K, M^\star, t^\star) = 1$; (ii) $(M^\star, t^\star) \notin Q$.[1]

We consider an even stronger definition where the adversary is additionally given access to a verification oracle $\mathsf{Verify}(K, ., .)$. At each verification query, the adversary chooses a pair (M, t) and the challenger returns the output of $\mathsf{Verify}(K, M, t) \in \{0, 1\}$. In this context, the adversary wins if one of these verification queries (M, t) satisfies: (i) $\mathsf{Verify}(K, M, t) = 1$; (ii) $(M, t) \notin Q$.

1. Show that the verification oracle does not make the adversary any stronger. Namely, any strongly unforgeable MAC remains strongly unforgeable when the adversary has a verification oracle.

Exercise 4. *CBC-MAC*

Prove that the following modifications of CBC-MAC (recalled in Figure 1) do not yield a secure fixed-length MAC:

1. Modify CBC-MAC so that a random IV (rather than $IV = \mathbf{0}$) is used each time a tag is computed (and the IV is output along with t_ℓ).

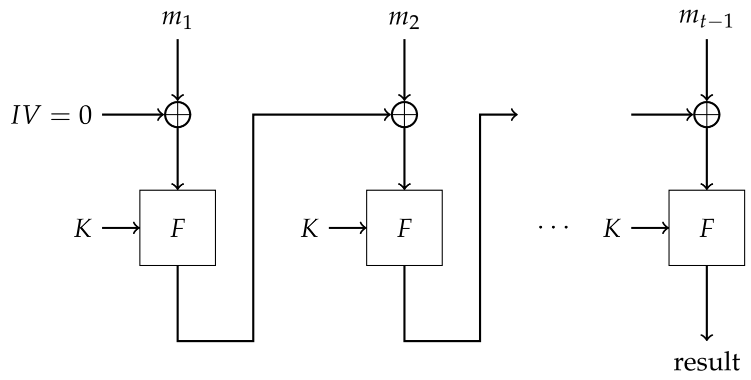

Figure 1: CBC-MAC

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

We now consider the following ECBC-MAC scheme, let $F : K \times X \to X$ be a PRP, 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: $m = m_1 | \ldots | m_{d-1} | (m_d \| \mathrm{pad}(m))$.

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

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 $m_0 \| \mathrm{pad}(m_0) \neq m_1 \| \mathrm{pad}(m_1)$. 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. Explain why the scheme is not secure if this padding does not add a new block if the message length is a multiple of the block length.

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.

[1] In the definition of **standard** unforgeability under chosen-message attacks, condition (ii) is replaced by $\forall (M_i, t_i) \in Q$, $M^\star \neq M_i$.

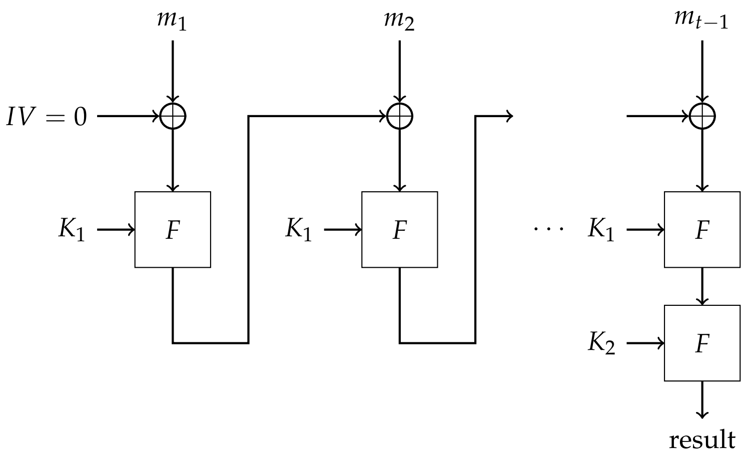

Figure 2: ECBC-MAC

Exercise 5. *Pseud-random synthetizers*

Let $n \in \mathbb{N}$ be a security parameter. Let $\mathbb{G}$ be a cyclic group of prime order $q > 2^n$ with a generator $g \in \mathbb{G}$. Recall that the Decisional Diffie-Hellman (DDH) assumption says that the following distributions

$$D_0 := \{(g^a, g^b, g^{ab}) \mid a, b \leftarrow U(\mathbb{Z}_q)\}, \qquad D_1 := \{(g^a, g^b, g^c) \mid a, b, c \leftarrow U(\mathbb{Z}_q)\}$$

are computationally indistinguishable.

A **synthesizer** $G : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{G}^{n \times n}$ is a length-squaring function which takes as input a random seed made of $2n$ scalars $\vec{a} = (a_1, \ldots, a_n) \leftarrow U(\mathbb{Z}_q^n)$, $\vec{b} = (b_1, \ldots, b_n) \leftarrow U(\mathbb{Z}_q^n)$ and outputs a $n \times n$ matrix

$$G\big((a_1, \ldots, a_n), (b_1, \ldots, b_n)\big) = \Big(g^{a_i b_j}\Big)_{i,j \in \{1,\ldots,n\}} = \begin{bmatrix} g^{a_1 b_1} & \ldots & g^{a_1 b_n} \\ g^{a_2 b_1} & \ldots & g^{a_2 b_n} \\ \vdots & \ddots & \vdots \\ g^{a_n b_1} & \ldots & g^{a_n b_n} \end{bmatrix}. \tag{1}$$

1. Show that an unbounded adversary (which can compute discrete logarithms in $\mathbb{G}$) can distinguish an output of G from a truly random matrix in $\mathbb{G}^{n \times n}$.

2. Show that $G : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{G}^{n \times n}$ is a pseudo-random generator under the DDH assumption in the group $\mathbb{G}$.

   Hint (but you may choose not to read it): Consider a sequence of n^2 hybrid experiments $\mathbf{Exp}_{k,\ell}$, for $k, \ell \in \{1, \ldots, n\}$, where the output of $G((a_1, \ldots, a_n), (b_1, \ldots, b_n))$ is replaced by a matrix of the form

   $$G^{(k,\ell)}\big((a_1, \ldots, a_n), (b_1, \ldots, b_n)\big) = \Big(g^{u_{ij}}\Big)_{i,j \in \{1,\ldots,n\}}$$

   where $u_{ij} = a_i b_j$ if $i > k$ or $(i = k) \wedge (j > \ell)$ and $u_{ij} \leftarrow U(\mathbb{Z}_q)$ otherwise. Define $G^{(0,0)}$ to be actual function of (1).