TD 4: CPA security and MACs

Exercise 1. [*Security of CBC mode*]

Suppose $\mathcal{E} = (E, D)$ is a block cipher defined over $(K, \mathcal{X})$ where $\mathcal{X} = \{0,1\}^n$. Let $N := |\mathcal{X}| = 2^n$. For any poly-bounded $l \geq 1$, we define a cipher $\mathcal{E}' = (E', D')$, with a key space $\mathcal{K}$, message space $\mathcal{X}^{\leq l}$, and ciphertext space $\mathcal{X}^{\leq l+1} \backslash \mathcal{X}^0$; that is the ciphertext space consists of all nonempty sequences of at most $l+1$ data blocks. Encryption and decryption are defined as follows.

Encryption

for $k \in \mathcal{K}$ and $m \in \mathcal{X}^{\leq l}$, with $v := |m|$, we define

$E'(k, m) :=$

compute $c \in \mathcal{X}^{v+1}$ as follows:

$c[0] \xleftarrow{R} (\mathcal{X})$

for $j \leftarrow 0$ to $v - 1$

$c[j+1] \leftarrow E(k, c[j]) \oplus m[j]$

output c;

Decryption

for $k \in \mathcal{K}$ and $c \in \mathcal{X}^{\leq l+1} \backslash \mathcal{X}^0$, with $v := |c| - 1$, we define

$D'(k, c) :=$

compute $m \in \mathcal{X}^v$ as follows:

for $j \leftarrow 0$ to $v - 1$ do

$m[j] \leftarrow D(k, c[j+1] \oplus c[j])$

output m;

1. Prove the correctness of the cipher.
2. Prove that if $\mathcal{E} = (E, D)$ is a secure block cipher defined over $(\mathcal{K}, \mathcal{X})$, and $N := |\mathcal{X}|$ is super-poly, then for any poly-bounded $l \geq 1$, the cipher $\mathcal{E}'$ described above is CPA-secure.

   In particular, for every CPA adversary $\mathcal{A}$ that attacks $\mathcal{E}'$ and makes at most Q queries to its challenger, there exists a BC (Block Cipher) adversary $\mathcal{B}$ that attacks $\mathcal{E}$, such that:

$$\mathrm{Adv}_{\mathcal{A}}^{CPA}(\mathcal{E}') \leq \frac{2Q^2 l^2}{N} + 2 \cdot \mathrm{Adv}_{\mathcal{B}}^{BC}(\mathcal{E})$$

Exercise 2. [*The malleability of CBC mode*]

Let c be the CBC encryption of some message $m \in \mathcal{X}^l$, where $\mathcal{X} := \{0,1\}^n$. You do not know m. Let $\Delta \in \mathcal{X}$. Show how to modify the ciphertext c to obtain a new ciphertext c' that decrypts to m', where $m'[0] = m[0] \oplus \Delta$, and $m'[i] = m[i]$ for $i = 1, \cdots, l-1$. That is, by modifying c appropriately, you can flip bits of your choice in the first block of the decryption of c without affecting any of the other blocks.

Definition 1. *A* MAC *system $\mathcal{I} = (S, V)$ is a pair of efficient algorithms, S and V, where*

- *S is a probabilistic (signing) algorithm, that given a key k, a message m, it produces a tag t where $t \xleftarrow{R} S(k, m)$.*
- *V is a deterministic (verification) algorithm that given a key k, a tag t, it outputs **accept** or **reject**.*
- *It requires correctness property: for all keys k and all messages m;*

$$Pr\{V(k, m, S(k, m)) = \textbf{\textit{accept}}\} = 1$$

Definition 2. *(**MAC security**) For a given MAC system $\mathcal{I} = (S, V)$, defined over $\mathcal{K}, \mathcal{M}, \mathcal{T}$, and a given adversary $\mathcal{A}$, the attack game runs as follows:*

- *The challenger picks a random $k \xleftarrow{R} \mathcal{K}$.*
- *$\mathcal{A}$ queries the challenger several times. For $i = 1, 2, \cdots$, the i^{th} signing query is a message $m_i \in \mathcal{M}$. Given m_i, the challenger computes a tag $t_i \xleftarrow{R} S(k, m_i)$, and then gives t_i to $\mathcal{A}$.*
- *$\mathcal{A}$ outputs a candidate forgery pair $(m, t) \in \mathcal{M} \times \mathcal{T}$ that is not among the signed pairs, i.e.,*

$$(m, t) \notin \{(m_1, t_1), (m_2, t_2), \cdots\}$$

We say that $\mathcal{A}$ wins *the above game if (m, t) is a valid pair under k. Moreover, we define:*

$$Advt_{\mathcal{A}}^{MAC}(\mathcal{I}) = Pr\{\mathcal{A}\ wins\}$$

Finally, $\mathcal{I}$ is a secure MAC*, if for all efficient adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible.*

Exercise 3. [*MAC and PRF*]

Recall that a pseudo-random function (PRF) defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ is an algorithm F that takes two inputs, a key $k \in \mathcal{K}$ and an input data block $x \in \mathcal{X}$, and outputs a value $y := F(k, x)$. For a PRF F, we define we define the deterministic MAC system $\mathcal{I} = (S, V)$ derived from F as:

- $S(k, m) := F(k, m)$
- $V(k, m, t) :=$ **accept** if $F(k, m) = t$, and **reject** otherwise

We note that a secure PRF implies a secure deterministic MAC (proof ignored).

1. 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 4. [*MAC 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 message 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 $(m, t) \notin Q$

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.

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