TD 5: Toward computational security of protocols

Margot Catinaud margot.catinaud@lmf.cnrs.fr
Théo Vignon theo.vignon@lmf.cnrs.fr

December 13, 2024

First off, let's start by define the primitive that we will use during all this exercise session, the symmetric encryption:

Definition: Symmetric encryption

A symmetric encryption scheme is defined by the following signature $(\mathcal{K}/2,\ \mathcal{E}/3,\ \mathcal{D}/2)$[a], where :

1. $\mathcal{K}$ is the *key generation function*, that given the security parameter η[b] and some random seed, generates the secret key used by the encryption and decryption functions.
2. $\mathcal{E}$ is the *encryption function*, that given the plaintext to encrypt, the key, and some randomness, outputs the associated ciphertext. We will also use the notation $\{m\}_{\mathsf{k}}^{r}$ for $\mathcal{E}(m, \mathsf{k}, r)$
3. $\mathcal{D}$ is the *decryption function*, that given a ciphertext and a key, outputs the associated plaintext or possibly a special error symbol $\perp$.

In this exercise session (and almost all the time), we assume that an encryption scheme is at least *correct*, meaning that the decryption of some encryption (with the same key) gives back the plaintext. Written formally,

$$\forall\ m\ r\ \mathsf{k}, \mathcal{D}(\{m\}_{\mathsf{k}}^{r}, \mathsf{k}) = m$$

[a] Here all those are interpreted as *deterministic* functions, to which we give their randomness explicity if necessary, we could also interpret them as probabilistic functions that follow some distributions.

[b] often, the security parameter will be given as argument as 1^η, the bitstring of length η only composed of 1. This is due to implicit assumptions.

(a) that the numbers are given in binary

(b) the functions are in polynomial time in their input

Therefore, we want the key generation function to be polynomial in η and not logarithmic in η

Exercise 1: $\texttt{CPA}$ and $\texttt{IND} - \texttt{CCA2}$

First, we need to adapt the definitions of $\texttt{IND} - \texttt{CPA}$ and $\texttt{IND} - \texttt{CCA2}$ for the symmetric encryption setting.

Definition: $\texttt{IND} - \texttt{CPA}$

Given a symmetric encryption scheme $(\mathcal{K},\ \mathcal{E},\ \mathcal{D})$. We define the $\texttt{IND} - \texttt{CPA}$ security notion as the indistinguishability between the following scenarios (the variations between those two scenarios are parametrized by a boolean b):

- First, we compute a key k from $\mathcal{K}$.
- Then, the adversary gets access to an oracle $\mathcal{O}_{\mathcal{E}}, \mathcal{O}^b_{\texttt{IND}-\texttt{CPA}}$ and returns a boolean b'. The oracles are defined as such (with r being some fresh randomness at every call)

$$\mathcal{O}_{\mathcal{E}}(m) := c = \{m\}^r_{\mathsf{k}}$$

$$\mathcal{O}^b_{\texttt{IND}-\texttt{CPA}}(m_0, m_1) = \begin{cases} \{m_b\}^r_{\mathsf{k}} & \text{if } |m_0| = |m_1| \\ \mathsf{witness} & \text{otherwise} \end{cases}$$

where $\mathsf{witness}$ is some special message. The adversary wins if $b' = b$
The advantage of an adversary $\mathcal{A}$ against the $\texttt{IND} - \texttt{CPA}$ game is

$$\mathbf{Adv}^{\eta}_{\texttt{IND}-\texttt{CPA}}(\mathcal{A}) = \left| \begin{array}{l} \mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^\eta, \mathsf{r_k}) : \mathcal{A}^{\mathcal{O}_{\mathcal{E}}, \mathcal{O}^0_{\texttt{IND}-\texttt{CPA}}}(1^\eta) = 0\right] \\ -\mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^\eta, \mathsf{r_k}) : \mathcal{A}^{\mathcal{O}_{\mathcal{E}}, \mathcal{O}^1_{\texttt{IND}-\texttt{CPA}}}(1^\eta) = 0\right] \end{array} \right|$$

Definition: $\texttt{IND} - \texttt{CCA2}$

Given a symmetric encryption scheme $(\mathcal{K},\ \mathcal{E},\ \mathcal{D})$. We define the $\texttt{IND} - \texttt{CCA2}$ security notion as the indistinguishability between the following scenarios (the variations between those two scenarios are parametrized by a boolean b):

- First, we compute a key k form $\mathcal{K}$, initialize a list log to the empty list $[\,]$
- Then, the adversary gets access to an oracle $\mathcal{O}_{\mathcal{E}}, \mathcal{O}_{\mathcal{D}}, \mathcal{O}^b_{\texttt{IND}-\texttt{CCA2}}$ and returns a boolean b'. The oracles are defined as such (with r being some fresh randomness at every call)

$$\mathcal{O}_{\mathcal{E}}(m) := c = \{m\}^r_{\mathsf{k}}$$

$$\mathcal{O}_{\mathcal{D}}(c) := \textbf{if } c \notin \mathsf{log} \textbf{ then } \mathcal{D}(c, \mathsf{k}) \textbf{ else } \mathsf{witness}$$

$$\mathcal{O}^b_{\texttt{IND}-\texttt{CCA2}}(m_0, m_1) = \begin{cases} \textbf{if } |m_0| = |m_1| \\ \textbf{then let } c = \{m_b\}^r_{\mathsf{k}} \textbf{ in } \mathbf{List}.\text{append}(c, \mathsf{log}); c \\ \textbf{else } \mathsf{witness} \end{cases}$$

where $\mathsf{witness}$ is some special message. The adversary wins if $b' = b$
The advantage of an adversary $\mathcal{A}$ against the $\texttt{IND} - \texttt{CPA}$ game is

$$\mathbf{Adv}^{\eta}_{\texttt{IND}-\texttt{CPA}}(\mathcal{A}) = \left| \begin{array}{l} \mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^\eta, \mathsf{r_k}) : \mathcal{A}^{\mathcal{O}_{\mathcal{E}}, \mathcal{O}_{\mathcal{D}}, \mathcal{O}^0_{\texttt{IND}-\texttt{CCA2}}}(1^\eta) = 0\right] \\ -\mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^\eta, \mathsf{r_k}) : \mathcal{A}^{\mathcal{O}_{\mathcal{E}}, \mathcal{O}_{\mathcal{D}}, \mathcal{O}^1_{\texttt{IND}-\texttt{CCA2}}}(1^\eta) = 0\right] \end{array} \right|$$

Let $\mathcal{S}$ be a symmetric encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$:

1. let $\mathcal{A}$ be an adversary against the $\texttt{IND} - \texttt{CPA}$ assumption, find an adversary $\mathcal{R}(\mathcal{A})$ (so, a reduction

Exercise 1: $\texttt{CPA}$ and $\texttt{IND}-\texttt{CCA2}$

built upon $\mathcal{A}$) such that if $\mathcal{A}$ succeeds to break the $\texttt{IND}-\texttt{CPA}$ assumption then $\mathcal{R}(\mathcal{A})$ also breaks the $\texttt{IND}-\texttt{CCA2}$ assumption.

2. Show that if an encryption scheme is $\texttt{IND}-\texttt{CCA2}$-secure then it is also $\texttt{IND}-\texttt{CPA}$-secure. Meaning, that you can upper-bound the advantage of an adversary against the $\texttt{IND}-\texttt{CPA}$ assumption by the advantage of an adversary against the $\texttt{IND}-\texttt{CCA2}$ assumption.

3. Do you think that $\texttt{IND}-\texttt{CPA}$ security implies $ICCAA$ security? Why?

Exercise 2: Integrity for symmetric encryption

We define here a new kind of security notion, the *integrity of ciphertext* ($\texttt{INT}-\texttt{CTXT}$). Intuitively, this notion capture that it is hard to make a valid ciphertext (ie that decrypts to an actual message) without the key.

Definition: $\texttt{INT}-\texttt{CTXT}$

Given a symmetric encryption scheme $(\mathcal{K},\ \mathcal{E},\ \mathcal{D})$. We define the $\texttt{INT}-\texttt{CTXT}$ security notion as the following scenario :

1. First, we compute a key k form $\mathcal{K}$, initialize a list log to the empty list $[\,]$, and some boolean win to false

2. Then, the adversary gets access to two oracles $\mathcal{O}_\mathcal{E}, \mathcal{O}_\mathcal{D}$ and finishes when it wants. The oracles are defined as such (with r being some fresh randomness at every call)

$$\mathcal{O}_\mathcal{E}(m) := \textbf{let } c = \{m\}_{\mathsf{k}}^{r} \textbf{ in } \textbf{List}.\text{append}(c, \mathsf{log});\ c$$

$$\mathcal{O}_\mathcal{D}(c) := \textbf{let } \mathsf{win} = \mathsf{win} \vee (\mathcal{D}(c, \mathsf{k}) \neq \bot \wedge c \notin \mathsf{log}) \textbf{ in } \mathsf{win}$$

The adversary wins if $\mathsf{win} = \mathsf{true}$.
The advantage of an adversary $\mathcal{A}$ against the $\texttt{INT}-\texttt{CTXT}$ game is

$$\mathbf{Adv}^{\eta}_{\texttt{INT}-\texttt{CTXT}}(\mathcal{A}) = \mathbf{Pr}\begin{bmatrix}\mathsf{k} \leftarrow \mathcal{K}(1^\eta, \mathsf{r_k}); \mathsf{log} \leftarrow [\,]; \\ \mathsf{win} \leftarrow \mathsf{false}; \mathcal{A}^{\mathcal{O}_\mathcal{E}, \mathcal{O}_\mathcal{D}}(1^\eta) : \mathsf{win} = \mathsf{true}\end{bmatrix}$$

Show that this definition is equivalent to the following definition

Exercise 2: Integrity for symmetric encryption

Definition: $\texttt{INT}-\texttt{CTXT}$ – Game-based setting

Given a symmetric encryption scheme $(\mathcal{K},\ \mathcal{E},\ \mathcal{D})$. We define the $\texttt{INT}-\texttt{CTXT}$ security notion as the indistinguishability between the following scenarios (the variations between those two scenarios are parametrized by a boolean b):

1. First, we compute a key k form $\mathcal{K}$, initialize a list log to the empty list $[\,]$
2. Then, the adversary gets access to two oracles $\mathcal{O}_{\mathcal{E}}, \mathcal{O}^b_{\mathcal{D}}$ and send back a boolean b'. The oracles are defined as such (with r being some fresh randomness at every call)

$$\mathcal{O}_{\mathcal{E}}(m) := \textbf{let } c = \{m\}^r_{\mathsf{k}} \textbf{ in } \textbf{List}.\text{append}(c, \mathsf{log});\ c$$

$$\mathcal{O}^b_{\mathcal{D}}(c) := \textbf{if } b \wedge c \notin \mathsf{log} \textbf{ then } \mathcal{D}(c, \mathsf{k}) \textbf{ else } \bot$$

The adversary wins if $b = b'$

The advantage of an adversary $\mathcal{A}$ against the $\texttt{INT}-\texttt{CTXT}$ game is

$$\mathbf{Adv}^{\eta}_{\texttt{INT}-\texttt{CTXT}}(\mathcal{A}) = \left| \begin{array}{l} \mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^\eta, \mathsf{r_k}); \mathsf{log} \leftarrow [\,] : \mathcal{A}^{\mathcal{O}_{\mathcal{E}}, \mathcal{O}^0_{\mathcal{D}}}(1^\eta) = 0\right] \\ -\mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^\eta, \mathsf{r_k}); \mathsf{log} \leftarrow [\,] : \mathcal{A}^{\mathcal{O}_{\mathcal{E}}, \mathcal{O}^1_{\mathcal{D}}}(1^\eta) = 0\right] \end{array} \right|$$

In the rest of the exercise session, we advise using the definition in the game-based setting.

Exercise 3: Integrity and privacy

Given a symmetric encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, show that if this encryption scheme is $\texttt{IND}-\texttt{CPA}$-secure and $\texttt{INT}-\texttt{CTXT}$-secure then it is $\texttt{IND}-\texttt{CCA2}$-secure.

Exercise 4: Wide-mouth frog

Recall the example of the *wide-mouth frog* protocol:

$$\begin{aligned} A \rightarrow S &: A, \{B, s, m_1\}_{\mathsf{k}_{AS}} \\ S \rightarrow B &: \{A, s, m_2\}_{\mathsf{k}_{BS}} \end{aligned}$$

In this exercise, we are interested in proving the strong secrecy of s in the computational model under the assumption that the symmetric encryption scheme used is both $\texttt{IND}-\texttt{CPA}$-secure and $\texttt{INT}-\texttt{CTXT}$-secure. To do that, we need to write down formally the protocol and what it means for an adversary to break the protocol.

Here, we will represent the protocol by a set of oracle $\{\mathsf{Init}, \mathsf{A}, \mathsf{S}, \mathsf{Register}\}$[a], we assume that among those oracles Init is a special oracle that is called at the beginning of the game and never after, as for the other oracles, the adversary can call them freely.

Here, we will represent our security property (the strong secrecy of s) as the indistinguishability of two sets of oracles for the adversary. Formally, we define the advantage of the adversary against sets of oracles as:

Exercise 4: Wide-mouth frog

Definition: Advantage of the adversary

Let $\{\mathcal{O}, \mathsf{Init}_{\mathcal{O}}\}, \{\mathcal{I}, \mathsf{Init}_{\mathcal{I}}\}$ be two sets of oracles. The advantage of an adversary $\mathcal{A}$ against the game $(\mathcal{O}, \mathcal{I})$ is :

$$\mathbf{Adv}^{\eta}_{(\mathcal{O},\mathcal{I})}(\mathcal{A}) = \left| \begin{array}{l} \quad\mathbf{Pr}\left[() \leftarrow \mathsf{Init}_{\mathcal{O}}() : \mathcal{A}^{\mathcal{O}}(1^{\eta}) = 0\right] \\ -\mathbf{Pr}\left[() \leftarrow \mathsf{Init}_{\mathcal{I}}() : \mathcal{A}^{\mathcal{I}}(1^{\eta}) = 0\right] \end{array} \right|$$

We can now formally define the oracle of the original wide-mouth frog protocol

Definition: Wide mouth frog

We define here the original version of the wide-mouth frog $\mathsf{WMF} := \{\mathsf{Init}_{\mathsf{WMF}}, \mathsf{A}_{\mathsf{WMF}}, \mathsf{S}_{\mathsf{WMF}}, \mathsf{Register}_{\mathsf{WMF}}\}$ as such:

```
$\mathsf{Init}_{\mathsf{WMF}}() :=$
    $\mathsf{k}_A \xleftarrow{\$} \mathsf{dkey};$
    $\mathsf{k}_B \xleftarrow{\$} \mathsf{dkey};$
    $\mathsf{keys} \leftarrow [\,];$
    $\mathsf{keys}[A] \leftarrow \mathsf{k}_A;$
    $\mathsf{keys}[B] \leftarrow \mathsf{k}_B.$

$\mathsf{A}_{\mathsf{WMF}}(I) :=$
    $\mathsf{s} \xleftarrow{\$} \mathsf{dkey};$
    $\mathsf{r} \xleftarrow{\$} \mathsf{rand};$
    $(A, \{I, \mathsf{s}, m_1\}^{\mathsf{r}}_{\mathsf{k}_A}).$
```

```
$\mathsf{S}_{\mathsf{WMF}}(x) :=$
    $\textbf{let } O, m = x \textbf{ in}$
    $\textbf{let } \mathsf{k}_{\mathcal{D}} = \mathsf{keys}[O] \textbf{ in}$
    $\textbf{let } I, \mathsf{s}, t = \mathcal{D}(m, \mathsf{k}_{\mathcal{D}}) \textbf{ in}$
    $\textbf{let } \mathsf{k}_{\mathcal{E}} = \mathsf{keys}[I] \textbf{ in}$
    $\textbf{if } t = m_1$
    $\textbf{then } \mathsf{r} \xleftarrow{\$} \mathsf{rand};$
        $\{O, \mathsf{s}, m_2\}^{\mathsf{r}}_{\mathsf{k}_{\mathcal{E}}}.$

$\mathsf{Register}_{\mathsf{WMF}}(x) :=$
    $\textbf{let } I, \mathsf{k} = x \textbf{ in}$
    $\textbf{if } \mathsf{keys}[I] = \ \textbf{then } \mathsf{keys}[I] \leftarrow \mathsf{k}.$
```

Once we have this formalization of the wide-mouth frog protocol, we still need to define our ideal game (the one where the strong secrecy of s holds for sure).

Exercise 4: Wide-mouth frog

Definition: Ideal game

We define here the idealized version of the wide mouth frog $\mathsf{Ideal} := \{\mathsf{Init}_{\mathsf{Ideal}}, \mathsf{A}_{\mathsf{Ideal}}, \mathsf{S}_{\mathsf{Ideal}}, \mathsf{Register}_{\mathsf{Ideal}}\}$ as such:

$$\mathsf{Init}_{\mathsf{Ideal}} := \mathsf{Init}_{\mathsf{WMF}},\ \mathsf{S}_{\mathsf{Ideal}} := \mathsf{S}_{\mathsf{WMF}},\ \mathsf{Register}_{\mathsf{Ideal}} := \mathsf{Register}_{\mathsf{WMF}}$$

and

```
$\mathsf{A}_{\mathsf{Ideal}}(I) :=$
    $\mathsf{s} \xleftarrow{\$} \mathsf{dkey};$
    $\mathsf{r} \xleftarrow{\$} \mathsf{rand};$
    if $I = B$
        then $(A, \{I, 0^{|\mathsf{s}|}, m_1\}^{\mathsf{r}}_{\mathsf{k}_A}).$
        else $(A, \{I, \mathsf{s}, m_1\}^{\mathsf{r}}_{\mathsf{k}_A}).$
```

So, our goal in this exercise, is to upper-bound $\mathbf{Adv}^{\eta}_{(\mathsf{WMF},\mathsf{Ideal})}(\mathcal{A})$ for any $\mathcal{A}$, with the advantage of some adversary (built on $\mathcal{A}$) against $\mathtt{IND}-\mathtt{CPA}$ and $\mathtt{INT}-\mathtt{CTXT}$. To do that, we will do it step by step, introducing multiple intermediary games. The first game that we want to introduce is the game that represents the application of $\mathtt{INT}-\mathtt{CTXT}$ assumption on the encryption scheme used by WMF:

Definition: WMF − CTXT

We define here the version of the WMF once the idealization of the $\mathtt{CTXT}$ was applied $\mathsf{WMF}-\mathtt{CTXT} := \{\mathsf{Init}_{\mathsf{WMF}-\mathtt{CTXT}}, \mathsf{A}_{\mathsf{WMF}-\mathtt{CTXT}}, \mathsf{S}_{\mathsf{WMF}-\mathtt{CTXT}}, \mathsf{Register}_{\mathsf{WMF}-\mathtt{CTXT}}\}$ as such:

```
$\mathsf{Init}_{\mathsf{WMF}-\mathtt{CTXT}}() :=$
    $() \leftarrow \mathsf{Init}_{\mathsf{WMF}};$
    $\mathsf{log} \leftarrow [\,].$

$\mathsf{A}_{\mathsf{WMF}-\mathtt{CTXT}}(I) :=$
    $\mathsf{s} \xleftarrow{\$} \mathsf{dkey};$
    $\mathsf{r} \xleftarrow{\$} \mathsf{rand};$
    let $c = \{I, \mathsf{s}, m_1\}^{\mathsf{r}}_{\mathsf{k}_A}$ in
    $\mathsf{log}[c, \mathsf{k}_A] \leftarrow (I, \mathsf{s}, m_1);$
    $(A, c).$
```

```
$\mathsf{S}_{\mathsf{WMF}-\mathtt{CTXT}}(x) :=$
    let $O, m = x$ in
    let $\mathsf{k}_{\mathcal{D}} = \mathsf{keys}[O]$ in
    let $I, \mathsf{s}, t = \mathsf{log}[m, \mathsf{k}_{\mathcal{D}}]$ in
    let $\mathsf{k}_{\mathcal{E}} = \mathsf{keys}[I]$ in
    if $t = m_1$
    then $\mathsf{r} \xleftarrow{\$} \mathsf{rand};$
        let $c = \{O, \mathsf{s}, m_2\}^{\mathsf{r}}_{\mathsf{k}_{\mathcal{E}}}$ in
        $\mathsf{log}[c, \mathsf{k}_{\mathcal{E}}] \leftarrow (O, \mathsf{s}, m_2);$
        $c.$

$\mathsf{Register}_{\mathsf{WMF}-\mathtt{CTXT}}(x) := \mathsf{Register}_{\mathsf{WMF}}(x)$
```

1. Let $\mathcal{A}$ be an adversary against the game $(\mathsf{WMF}, \mathsf{WMF}-\mathtt{CTXT})$, find an adversary $\mathcal{R}(\mathcal{A})$ (so, a reduction built upon $\mathcal{A}$) such that if $\mathcal{A}$ wins against the game $(\mathsf{WMF}, \mathsf{WMF}-\mathtt{CTXT})$ then $\mathcal{R}(\mathcal{A})$ breaks the $\mathtt{INT}-\mathtt{CTXT}$ assumption.

Exercise 4: Wide-mouth frog

2. Give (for any $\mathcal{A}$) an upper-bound of $\mathbf{Adv}^{\eta}_{(\mathsf{WMF},\mathsf{WMF}-\mathtt{CTXT})}(\mathcal{A})$ as a function of the advantage of a adversary against the $\mathtt{INT}-\mathtt{CTXT}$ assumption.
3. Define $\mathsf{WMF}-\mathtt{CPA}$, the set of oracles that correspond to $\mathsf{WMF}-CTXT$ after the application of the $\mathtt{IND}-\mathtt{CPA}$ assumption.
4. Give (for any $\mathcal{A}$) an upper bound of $\mathbf{Adv}^{\eta}_{(\mathsf{WMF}-\mathtt{CTXT},\mathsf{WMF}-\mathtt{CPA})}(\mathcal{A})$ as a function of the advantage of an adversary against the $\mathtt{IND}-\mathtt{CPA}$ assumption.
5. Define $\mathsf{Ideal}-\mathtt{CTXT}$, the set of oracles that correspond to Ideal after the application of the $\mathtt{INT}-\mathtt{CTXT}$ assumption.
6. Give (for any $\mathcal{A}$) an upper bound of $\mathbf{Adv}^{\eta}_{(\mathsf{Ideal},\mathsf{Ideal}-\mathtt{CTXT})}(\mathcal{A})$ as a function of the advantage of an adversary against the $\mathtt{INT}-\mathtt{CTXT}$ assumption.
7. Define $\mathsf{Ideal}-\mathtt{CPA}$, the set of oracles that correspond to $\mathsf{Ideal}-\mathtt{CTXT}$ after the application of the $\mathtt{IND}-\mathtt{CPA}$ assumption.
8. Give (for any $\mathcal{A}$) an upper bound of $\mathbf{Adv}^{\eta}_{(\mathsf{Ideal}-\mathtt{CTXT},\mathsf{Ideal}-\mathtt{CPA})}(\mathcal{A})$ as a function of the advantage of an adversary against the $\mathtt{IND}-\mathtt{CPA}$ assumption.
9. Show that for all $\mathcal{A}$ $\mathbf{Adv}^{\eta}_{(\mathsf{WMF}-\mathtt{CPA},\mathsf{Ideal}-\mathtt{CPA})}(\mathcal{A})=0$
10. Using all the previous questions, give for any $\mathcal{A}$ an upper bound of $\mathbf{Adv}^{\eta}_{(\mathsf{WMF},\mathsf{Ideal})}(\mathcal{A})$ as a function of the advantage of (multiples) adversary against the $\mathtt{IND}-\mathtt{CPA}$ and $\mathtt{INT}-\mathtt{CTXT}$ assumption.

[a] The number, inputs, name of the oracles are specific to the exact protocol at had here, but the method is generic.

Exercise 5: MAC and encryption

One last question that remains is how to build a symmetric encryption scheme that is both $\mathtt{IND}-\mathtt{CPA}$ and $\mathtt{INT}-\mathtt{CTXT}$. This is exactly the goal of this exercise. To do that, we want to introduce some building block (and also of independent interest), the message authentication code scheme :

Definition: Message authentication code

A message authentication code (MAC) scheme is defined by the following signature ($\mathcal{K}/2$, $\mathcal{S}/2$, $\mathcal{V}/3$)[a], where :

1. $\mathcal{K}$ is the *key generation function*, that given the security parameter η and some random seed, generates the secret key used by the signature and verification functions.
2. $\mathcal{S}$ is the *signing function*, that given the message to sign and the key, outputs the associated signature.
3. $\mathcal{V}$ is the *verification function*, that given a message, a signature and, a key, either *accepts* (outputs the boolean true) or *refuses* (outputs the boolean false) the signature.

In this exercise session (and almost all the time), we assume that a message authentication code scheme is at least *correct*, meaning that the verification of a signature of the same message under the same key accepts. Written formally,

$$\forall\ m\ \mathsf{k}, \mathcal{V}(m, \mathcal{S}(m,\mathsf{k}),\mathsf{k}) = \mathsf{true}$$

[a] all those are interpreted as *deterministic* functions as before, with the same remark

Exercise 5: MAC and encryption

Here, from an $\mathtt{IND-CPA}$ symmetric encryption scheme $(\mathcal{K},\ \mathcal{E},\ \mathcal{D})$, and a $\mathtt{SUF-MAC}$ MAC scheme $(\mathcal{K}_{\mathcal{S}},\ \mathcal{S},\ \mathcal{V})$, we want to find a way to build a symmetric encryption that is both $\mathtt{IND-CPA}$ and $\mathtt{INT-CTXT}$. First off, let's define the $\mathtt{SUF-MAC}$ security notion for MAC (We give the usual definition as well as the game-based version[a]):

Definition: $\mathtt{SUF-MAC}$

Given a MAC scheme $(\mathcal{K}_{\mathcal{M}},\ \mathcal{S},\ \mathcal{V})$. We define the $\mathtt{SUF-MAC}$ security notion as the following scenario :

1. First, we compute a key k form $\mathcal{K}_{\mathcal{M}}$, initialize a list log to the empty list $[\,]$, and some boolean win to false
2. Then, the adversary gets access to two oracles $\mathcal{O}_{\mathcal{S}}, \mathcal{O}_{\mathcal{V}}$ and finishes when it wants. The oracles are defined as such

$$\mathcal{O}_{\mathcal{S}}(m) := \textbf{let } s = \mathcal{S}(m, \mathsf{k}) \textbf{ in } \textbf{List}.\text{append}((s, m), \mathsf{log});\ s$$

$$\mathcal{O}_{\mathcal{S}}(s, m) := \textbf{let } \mathsf{win} = \mathsf{win} \vee (\mathcal{V}(m, s, \mathsf{k}) \wedge (s, m) \notin \mathsf{log}) \textbf{ in } \mathsf{win}$$

The adversary wins if $\mathsf{win} = \mathsf{true}$.
The advantage of an adversary $\mathcal{A}$ against the $\mathtt{SUF-MAC}$ game is

$$\mathbf{Adv}^{\eta}_{\mathtt{SUF-MAC}}(\mathcal{A}) = \mathbf{Pr}\begin{bmatrix}\mathsf{k} \leftarrow \mathcal{K}_{\mathcal{M}}(1^{\eta}, \mathsf{r_k}); \mathsf{log} \leftarrow [\,]; \\ \mathsf{win} \leftarrow \mathsf{false}; \mathcal{A}^{\mathcal{O}_{\mathcal{S}}, \mathcal{O}_{\mathcal{V}}}(1^{\eta}) : \mathsf{win} = \mathsf{true}\end{bmatrix}$$

Definition: $\mathtt{SUF-MAC}$ – Game-based setting

Given a symmetric encryption scheme $(\mathcal{K}_{\mathcal{M}},\ \mathcal{S},\ \mathcal{V})$. We define the $\mathtt{SUF-MAC}$ security notion as the indistinguishability between the following scenarios (the variations between those two scenarios are parametrized by a boolean b):

1. First, we compute a key k form $\mathcal{K}$, initialize a list log to the empty list $[\,]$
2. Then, the adversary gets access to two oracles $\mathcal{O}_{\mathcal{S}}, \mathcal{O}^{b}_{\mathcal{V}}$ and sends back a boolean b'. The oracles are defined as such

$$\mathcal{O}_{\mathcal{S}}(m) := \textbf{let } s = \mathcal{S}(m, \mathsf{k}) \textbf{ in } \textbf{List}.\text{append}((s, m), \mathsf{log});\ c$$

$$\mathcal{O}^{b}_{\mathcal{V}}(m, s) := \textbf{if } b \vee (s, m) \notin \mathsf{log} \textbf{ then } \mathcal{V}(m, s, \mathsf{k})\textbf{else } \mathsf{false}$$

The adversary wins if $b = b'$

The advantage of an adversary $\mathcal{A}$ against the $\mathtt{SUF-MAC}$ game is

$$\mathbf{Adv}^{\eta}_{\mathtt{SUF-MAC}}(\mathcal{A}) = \left|\begin{array}{l}\mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^{\eta}, \mathsf{r_k}); \mathsf{log} \leftarrow [\,] : \mathcal{A}^{\mathcal{O}_{\mathcal{S}}, \mathcal{O}^{0}_{\mathcal{V}}}(1^{\eta}) = 0\right] \\ -\mathbf{Pr}\left[\mathsf{k} \leftarrow \mathcal{K}(1^{\eta}, \mathsf{r_k}); \mathsf{log} \leftarrow [\,] : \mathcal{A}^{\mathcal{O}_{\mathcal{S}}, \mathcal{O}^{1}_{\mathcal{V}}}(1^{\eta}) = 0\right]\end{array}\right|$$

First, to build an encryption scheme, we must define the key generation function. Here in all the way to combine the two schemes, we will use the same construction for the key generation function, so we will define it once and for all,

Exercise 5: MAC and encryption

Definition: key generation function

Given $\mathcal{K}$ the key generation function of the encryption scheme and $\mathcal{K}_{\mathcal{M}}$ the key generation function of the MAC scheme. We define $\mathcal{K}_{\cup}$ as :

$$\mathcal{K}_{\cup}(1^{\eta}, (\mathsf{r}_1, \mathsf{r}_2)) := (\mathcal{K}(1^{\eta}, \mathsf{r}_1), \mathcal{K}_{\mathcal{M}}(1^{\eta}, \mathsf{r}_2))$$

A first way to combine the two scheme is to encrypt and mac at the same time:

Definition: Encrypt-and-MAC

We define a new symmetric encryption scheme form the previous two $(\mathcal{K}_{\mathtt{E\&M}}, \mathcal{E}_{\mathtt{E\&M}}, \mathcal{D}_{\mathtt{E\&M}})$. As we already mention, $\mathcal{K}_{\mathtt{E\&M}} := \mathcal{K}_{\cup}$. As for the encryption:

```
$\mathcal{E}_{\mathtt{E\&M}}(m, (\mathsf{k}_1, \mathsf{k}_2), r) :=$
  let $c = \mathcal{E}(m, \mathsf{k}_1, r)$ in
  let $s = \mathcal{S}(m, \mathsf{k}_2)$ in
    $(c, s)$
```

1. Define the associated decryption function
2. Show that this encryption scheme is not $\mathtt{IND} - \mathtt{CPA}$ secure.
3. Show that this encryption scheme is not $\mathtt{INT} - \mathtt{CTXT}$ secure.

A second way, could be to mac then encrypt.

Definition: MAC-then-encrypt

We define a new symmetric encryption scheme form the previous two $(\mathcal{K}_{\mathtt{MtE}}, \mathcal{E}_{\mathtt{MtE}}, \mathcal{D}_{\mathtt{MtE}})$. As we already mention, $\mathcal{K}_{\mathtt{MtE}} := \mathcal{K}_{\cup}$. As for the others two:

```
$\mathcal{E}_{\mathtt{MtE}}(m, (\mathsf{k}_1, \mathsf{k}_2), r) :=$
  let $s = \mathcal{S}(m, \mathsf{k}_2)$ in
    $\mathcal{E}((m, s), \mathsf{k}_1, r)$
```

```
$\mathcal{D}_{\mathtt{MtE}}(c, (\mathsf{k}_1, \mathsf{k}_2)) :=$
  let $(m, s) = \mathcal{D}(c, \mathsf{k}_1)$ in
  if $\mathcal{S}(m, s, \mathsf{k}_2)$
    then $m$ else $\perp$
```

4. Show that this encryption scheme is $\mathtt{IND} - \mathtt{CPA}$ secure.
5. Show that this encryption scheme is not $\mathtt{INT} - \mathtt{CTXT}$ secure.

One last way to do it, would be to encrypt then mac.

Exercise 5: MAC and encryption

Definition: Encrypt-then-MAC

We define a new symmetric encryption scheme form the previous two $(\mathcal{K}_{\mathtt{EtM}}, \mathcal{E}_{\mathtt{EtM}}, \mathcal{D}_{\mathtt{EtM}})$. As we already mention, $\mathcal{K}_{\mathtt{EtM}} := \mathcal{K}_{\cup}$. As for the others two:

```
$\mathcal{E}_{\mathtt{EtM}}(m, (\mathsf{k}_1, \mathsf{k}_2), r) :=$
  let $c = \mathcal{E}(m, \mathsf{k}_1, r)$ in
  let $s = \mathcal{S}(c, \mathsf{k}_2)$ in
    $(c, s)$
```

```
$\mathcal{D}_{\mathtt{EtM}}((c, s), (\mathsf{k}_1, \mathsf{k}_2)) :=$
  let $m = \mathcal{D}(c, \mathsf{k}_1)$ in
  if $\mathcal{S}(c, s, \mathsf{k}_2)$
    then $m$ else $\perp$
```

6. Show that this encryption scheme is $\mathtt{IND} - \mathtt{CPA}$ secure.

7. Show that this encryption scheme is $\mathtt{INT} - \mathtt{CTXT}$ secure.

[a] you can show that they are equivalent if you want, but it is very similar to the first exercise