CS-E4340 Cryptography: Exercise Sheet 4

—Message Authentication Codes (MACs) & Pseudorandom Functions (PRFs)—

Submission Deadline: October 3, 11:30 via MyCourses

Each exercise can give up to two participation points, 2 for a mostly correct solution and 1 point for a good attempt. Overall, the exercise sheet gives at most 4 participation points.

Exercise Sheet 4 is intended to help...

(a) ...understand the *definition* of a message authentication code (MAC).
(b) ...understand the *relation* between PRFs and MACs.
(c) ...understand and practice how to define the security for a cryptographic primitive via security notions.

Exercise 1 shows that a UNF-CMA-secure MAC might leak the message it authenticates.

Exercise 2 explores the differences/similarities between one-way functions and UNF-CMA-secure MAC schemes.

Exercise 3 constructs variants of MAC schemes and the goal is to distinguish modifications which harm security from modifications which don't harm security.

Exercise 4 deepens understanding of the UNF-CMA game and provide some definitions in which no scheme can be secure. In this case, we say that these are variants of the MAC game which are *trivial to break* (trivial does not mean that the exercise is easy. Rather, it means that the constructions are not meaningful.).

Exercise 5 is an advanced counterexample for showing that not every UNF-CMA-secure MAC is a one-way function. (In fact, every UNF-CMA-secure MAC is a so-called *distributional* one-way function. Ask Miikka Tiainen if you are curious on the definition and its implications.)

Hint: *Lecture 4* and the beginning of *Lecture 5* cover message authentication codes, so you can have a look at both lecture videos to help with this exercise sheet.

Exercise 1 (MACs can leak the message)**.** Let m_1 be a UNF-CMA secure MAC scheme. Prove that also m_2 is a UNF-CMA secure MAC scheme, where

$$m_2.\mathsf{mac}(k,x) := m_1.\mathsf{mac}(k,x)||x$$

and

```
$m_2.\mathsf{ver}(k,x,t)$
**assert** $t \neq \bot$
$t' \leftarrow t_{1...|t|-|x|}$
$x' \leftarrow t_{|t|-|x|+1...|t|}$
**if** $x' \neq x$
    **return** $0$
**return** $m_1.\mathsf{ver}(k,x,t')$
```

Hint: You need to provide a reduction in pseudo-code (main task) and show that the reduction works. In order to show that the reduction works as it should (explain in your solution what this means), you can either provide the main conceptual argument (in text) or an inlining proof (in pseudocode) as in the lecture.

Exercise 2 (OWFs & MACs)**.** Prove or disprove: For all one-way functions f, m is an UNF-CMA secure MAC scheme, where $m.\mathsf{mac}(k,x) = f(k||x)$ and

```
$m.\mathsf{ver}(k,x,t)$
$t' \leftarrow f(k||x)$
**if** $t' \neq t$
    **return** $0$
**else**
    **return** $1$
```

Hint: If you believe that this is an UNF-CMA secure MAC scheme, see hint for Exercise 1. If you believe that this is not necessarily an UNF-CMA secure MAC scheme, define a counterexample OWF f and provide an adversary in pseudocode. See lecture notes for Lecture 4 for an example on how to write such adversary pseudocode.

Exercise 3 (Candidate MAC schemes)**.** Let f be a secure (λ, λ)-PRF. Consider the four MAC schemes m_1, m_2, m_3, m_4 given below.

1. Which of these MAC-schemes are UNF-CMA and which are not? Justify your intuition.
2. Choose one of the MAC schemes m_i that you think is not UNF-CMA secure. Provide an adversary $\mathcal{A}$ (in pseudocode) that can distinguish between the real and ideal games $\mathtt{Gunf\text{-}cma}^b_m$ (see Chapter 3 in the Crypto Companion). An intuitive explanation for why the adversary works suffices in this exercise.

Note: Below, the syntax $x[i]$ refers to the value assigned to x during the ith loop. Moreover, $y[i] \leftarrow f(k, x[i])$ is the output value of f during the ith loop.

```
$m_1.\mathsf{mac}(k, x)$
$j \leftarrow \lambda - (|x| \bmod \lambda)$
$x' \leftarrow x||0^j,\ h \leftarrow \frac{|x'|}{\lambda}$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    $y[i] \leftarrow f(k, x[i])$
$t \leftarrow (y[0], y[1], ..., y[h-1])$
return $t$
```

```
$m_2.\mathsf{mac}(k, x)$
$j \leftarrow \lambda - (|x| \bmod \lambda)$
$x' \leftarrow x||0^j,\ h \leftarrow \frac{|x'|}{\lambda}$
$y[-1] \leftarrow 0^\lambda$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    $y[i] \leftarrow f(k, x[i] \oplus y[i-1])$
$t \leftarrow (y[0], y[1], ..., y[h-1])$
return $t$
```

```
$m_3.\mathsf{mac}(k, x)$
$j \leftarrow \lambda - (|x| + 1) \bmod \lambda)$
$x' \leftarrow x||1||0^j,\ h \leftarrow \frac{|x'|}{\lambda}$
$y[-1] \leftarrow 0^\lambda$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    $y[i] \leftarrow f(k, x[i] \oplus y[i-1])$
$t \leftarrow (y[0], y[1], ..., y[h-1])$
return $t$
```

```
$m_4.\mathsf{mac}(k, x)$
$j \leftarrow \lambda - ((|x| + 1) \bmod \lambda)$
$x' \leftarrow x||1||0^j,\ h \leftarrow \frac{|x'|}{\lambda}$
$y[-1] \leftarrow 0^\lambda$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    $y[i] \leftarrow f(k, x[i] \oplus y[i-1])$
$t \leftarrow y[h-1]$
return $t$
```

```
$m_1.\mathsf{ver}(k, x, t)$
$j \leftarrow \lambda - (|x| \bmod \lambda)$
$x' \leftarrow x||0^j,\ h \leftarrow \frac{|x'|}{\lambda}$
$(y[0], y[1], ..., y[h-1]) \leftarrow t$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    if $y[i] \neq f(k, x[i])$
        return $0$
return $1$
```

```
$m_2.\mathsf{ver}(k, x, t)$
$j \leftarrow \lambda - (|x| \bmod \lambda)$
$x' \leftarrow x||0^j,\ h \leftarrow \frac{|x'|}{\lambda}$
$y[-1] \leftarrow 0^\lambda$
$(y[0], y[1], ..., y[h-1]) \leftarrow t$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    if $y[i] \neq f(k, x[i] \oplus y[i-1])$
        return $0$
return $1$
```

```
$m_3.\mathsf{ver}(k, x, t)$
$z \leftarrow \lambda - ((|x| + 1) \bmod \lambda)$
$x' \leftarrow x||1||0^z,\ h \leftarrow \frac{|x'|}{\lambda}$
$y[-1] \leftarrow 0^\lambda$
$(y[0], ..., y[h-1]) \leftarrow t$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    if $y[i] \neq f(k, x[i] \oplus y[i-1])$
        return $0$
return $1$
```

```
$m_4.\mathsf{ver}(k, x, t)$
$z \leftarrow \lambda - (|x| + 1) \bmod \lambda)$
$x' \leftarrow x||1||0^z,\ h \leftarrow \frac{|x'|}{\lambda}$
$y[-1] \leftarrow 0^\lambda$
for $i = 0$ until $h - 1$
    $\ell \leftarrow \lambda i + 1$
    $r \leftarrow \lambda i + \lambda$
    $x[i] \leftarrow x'_{\ell..r}$
    $y[i] \leftarrow f(k, x[i] \oplus y[i-1])$
if $y\left[\frac{|x'|}{\lambda} - 1\right] = t$ :
    return $1$
else return $0$
```

Exercise 4. (Security Models, Weak Unforgeability) In this exercise, we aim to understand what would be a good model for a weak unforgeability definition that captures that only the message m is authenticated whereas the tag might be malleable. We capture weak unforgeability as computational indistinguishability between a real game $\texttt{Gwunf-cma}^0_m$ and an ideal game $\texttt{Gwunf-cma}^1_m$. We define the real game for weak unforgeability under chosen message attacks (wUNF-CMA) as the real game for UNF-CMA security, i.e., $\texttt{Gwunf-cma}^0_m := \texttt{Gunf-cma}^0_m$, where m is a message authentication scheme. Below, we give three candidates for the ideal game $\texttt{Gwunf-cma}^1_m$. You need to choose one candidate such that weak unforgeability (integrity on the message only) is best captured. Justify your choice.

```
Package Parameters
$\lambda$ : security parameter
$m$ : MAC scheme

Package State
$k$ : k
$\mathcal{L}$ : list

$\mathsf{MAC}(x)$
if $k = \bot$
  $k \leftarrow\!\!\$\ \{0,1\}^\lambda$
$t \leftarrow m.\mathsf{mac}(k, x)$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{x\}$
return $t$

$\mathsf{VERIFY}(x, t)$
assert  $x \neq \bot$
if $(x) \in \mathcal{L}$ and if $m.\mathsf{ver}(k, x, t) = 1$
  return 1
else  return 0
```

```
Package Parameters
$\lambda$ : security parameter
$m$ : MAC scheme

Package State
$k$ : k
$\mathcal{L}$ : list

$\mathsf{MAC}(x)$
if $k = \bot$
  $k \leftarrow\!\!\$\ \{0,1\}^\lambda$
$t \leftarrow m.\mathsf{mac}(k, x)$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{x\}$
return $t$

$\mathsf{VERIFY}(x, t)$
assert  $x \neq \bot$
if $(x) \in \mathcal{L}$
  return 1
else  return 0
```

```
Package Parameters
$\lambda$ : security parameter
$m$ : MAC scheme

Package State
$k$ : k
$\mathcal{L}$ : list

$\mathsf{MAC}(x)$
if $k = \bot$
  $k \leftarrow\!\!\$\ \{0,1\}^\lambda$
$t \leftarrow m.\mathsf{mac}(k, x)$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, t)\}$
return $t$

$\mathsf{VERIFY}(x, t)$
assert  $x \neq \bot$
if $(x, t) \in \mathcal{L}$
  return 1
else  return 0
```

Exercise 5. (Advanced counterexamples) Assume the existence of secure $(*, \lambda)$-PRFs. Prove that there exists an UNF-CMA-secure MAC scheme m such that the function

$$f_m : z \mapsto x||m.\mathsf{mac}(k, x) \text{ where } k = z_{1..\ell} \text{ and } x = z_{\ell+1..|x|} \text{ for } \ell = \left\lceil \frac{|z|}{2} \right\rceil$$

is not a one-way function. Notation: k is the first half of the input of f_m, and x is the second half of the input of f_m. When the input is of odd length, then x has one bit more than k.

Hint: Consult the counterexample theorems in the Crypto Companion.