Applied Cryptography, FS2023
Mar 9–10, 2023 // Version 1.2

Exercise Sheet 3

Prof. Kenny Paterson, F. Günther
M. Backendal, M. Filić

Group Discussion (CTR mode and deterministic decryption). Take some time to think about and discuss:

(a) The lectures mentioned different methods for selecting the starting counter *ctr* in CTR mode. Why is picking *ctr* carefully important? Which were the methods and what were their advantages and disadvantages?

(b) In the definition of symmetric encryption schemes, why not let the decryption algorithm Dec be probabilistic?

Suggested focus. Attempt these problems
before class: Problem 1, Problem 2 and Problem 3 part (a).
in class: Problem 3 parts (b) and (c), Problem 4.
in your own time: Problem 3 parts (d) and (e).

Suggested reading. Reading the following sections in the Boneh-Shoup book [1] might help with the problems on this exercise sheet: Sections 5.4.4 (Padding), 5.3 (IND-CPA security) and 4.4 (PRFs). You might also be interested in Sections 5.4.2 and 5.4.3 (Randomized CTR mode and CBC mode).

Problem 1 (Padding). Block ciphers encrypt *blocks* of bits, e.g. n bits at a time. Therefore, if a message has a length that is not a multiple of the block size n, it might need some pre-processing before it can be encrypted with a block cipher mode of operation such as CBC. One common way of handling this issue is to *pad* the message to a length that is a multiple of n. In this exercise, we look at different ways of padding a message by appending special padding bits to the end of the message. We ask whether the padding is *reversible*. That is, for any message, the question is whether the original message can be recovered after padding it to a multiple of the block size. Determine for each of the following padding schemes if they are reversible or not. If yes, explain how to recover the message and why recovery is guaranteed to succeed. If not, give an example of a message for which recovery fails.

(a) **Null Padding:** Append 0's to the message until it is a multiple of the block size.

(b) **Bit Padding:** Let N be the number of bits necessary to pad a message to a multiple of the block length. Append the string 10^{N-1}, that is, a 1 followed by $N-1$ 0's, to the message. To ensure that $N > 0$, if the message is already a multiple of the block size, let N be the block length (i.e. let $N = n$) and append a full block.

(c) **PKCS7 Padding:** Assume the message is an integer number of bytes, but not an integer number of blocks. Let N be the number of bytes necessary to pad to a multiple of the block length. To ensure that $N > 0$, if the message is already a multiple of the block length, let N be the block length (in bytes). Now pad with N bytes, each byte set to the value N. For example, if $N = 3$, append 3 bytes to the message, each byte set to 00000011.

(d) **PKCS7 Variant:** This padding scheme works exactly like PKCS7 padding, except that if the message is already a multiple of the block length, then it does not add any padding.

(e) **Discussion:** Some of the padding schemes above add a whole extra block if the message is already a multiple of the block length, whereas others do not. Adding a superfluous block of course has negative impact on efficiency. Based on the examples you have seen, do you think length extension is necessary for a reversible padding scheme?

Problem 2 (Block cipher modes of operation). Let $\mathsf{E}\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher that is PRP-secure. Below are several block cipher modes of operation (applied to E), each producing a symmetric encryption scheme SE = (KGen, Enc, Dec) for KGen that samples and returns a key uniformly at random from $\{0,1\}^k$. For each of the modes: (1) describe the corresponding decryption procedure Dec, and (2) show that the resulting symmetric encryption scheme SE does not have IND-CPA security (that means describe an adversary and compute its advantage). For simplicity, our symmetric encryption schemes are designed to work only with messages and ciphertexts of any length that is a multiple of the block length n.

In order to solve this problem, it might be helpful to draw a diagram showing how each of the encryption algorithms work.

(a) $\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}}(\mathcal{A}_a) = \ldots$

```
Enc$(\mathsf{K}, m_1 \| \ldots \| m_\ell)$
$r_0 \leftarrow_{\$} \{0,1\}^n$ ; $c_0 \leftarrow r_0$
For $i = 1, \ldots, \ell$ do
    $r_i \leftarrow \mathsf{E}(\mathsf{K}, m_i)$
    $c_i \leftarrow r_i \oplus r_{i-1}$
Return $c_0 \| \ldots \| c_\ell$
```

```
Dec$(\mathsf{K}, c_0 \| \ldots \| c_\ell)$
```

```
Adversary $\mathcal{A}_a^{\mathrm{LoR}}$
```

(b) $\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}}(\mathcal{A}_b) = \ldots$

```
Enc$(\mathsf{K}, m_1 \| \ldots \| m_\ell)$
$c_0 \leftarrow_{\$} \{0,1\}^n$
For $i = 1, \ldots, \ell$ do
    $c_i \leftarrow \mathsf{E}(\mathsf{K}, m_i) \oplus c_{i-1}$
Return $c_0 \| \ldots \| c_\ell$
```

```
Dec$(\mathsf{K}, c_0 \| \ldots \| c_\ell)$
```

```
Adversary $\mathcal{A}_b^{\mathrm{LoR}}$
```

(c) $\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}}(\mathcal{A}_c) = \ldots$

```
Enc$(\mathsf{K}, m_1 \| \ldots \| m_\ell)$
$c_0 \leftarrow_{\$} \{0,1\}^n$ ; $m_0 \leftarrow c_0$
For $i = 1, \ldots, \ell$ do
    $c_i \leftarrow \mathsf{E}(\mathsf{K}, m_i) \oplus m_{i-1}$
Return $c_0 \| \ldots \| c_\ell$
```

```
Dec$(\mathsf{K}, c_0 \| \ldots \| c_\ell)$
```

```
Adversary $\mathcal{A}_c^{\mathrm{LoR}}$
```

(d) $\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}}(\mathcal{A}_d) = \ldots$

$\underline{\mathrm{Enc}(\mathsf{K}, m_1 \,\|\, \ldots \,\|\, m_\ell)}$	$\underline{\mathrm{Dec}(\mathsf{K}, c_0 \,\|\, \ldots \,\|\, c_\ell)}$	Adversary $\mathcal{A}_d^{\mathrm{LoR}}$
$c_0 \leftarrow_\$ \{0,1\}^n$; $r_0 \leftarrow c_0$ For $i = 1, \ldots, \ell$ do $r_i \leftarrow r_{i-1} \oplus m_i$ $c_i \leftarrow \mathsf{E}(\mathsf{K}, r_i)$ Return $c_0 \,\|\, \ldots \,\|\, c_\ell$		

(e) $\mathbf{Adv}_{\mathrm{SE}}^{\mathrm{IND\text{-}CPA}}(\mathcal{A}_e) = \ldots$

$\underline{\mathrm{Enc}(\mathsf{K}, m_1 \,\|\, \ldots \,\|\, m_\ell)}$	$\underline{\mathrm{Dec}(\mathsf{K}, c_0 \,\|\, \ldots \,\|\, c_\ell)}$	Adversary $\mathcal{A}_e^{\mathrm{LoR}}$
$c_0 \leftarrow \mathsf{E}(\mathsf{K}, m_1)$ For $i = 1, \ldots, \ell$ do $c_i \leftarrow \mathsf{E}(\mathsf{K}, c_{i-1} \oplus m_i)$ Return $c_0 \,\|\, \ldots \,\|\, c_\ell$		

Mode (a) is similar to CBC, except the XOR happens after, rather than before, the block cipher application. Mode (c) is essentially the same as CBC decryption. Mode (e) is similar to CBC, except the IV is not randomized.

Problem 3 (IND-CPA security). Suppose SE $=$ (KGen, Enc, Dec) is an IND-CPA secure encryption scheme with key space $\mathcal{K}$ and message space $\mathcal{M}$, such that $\mathcal{K} = \mathcal{M} = \{0,1\}^n$ for some even integer n. Which of the following encryption algorithms are guaranteed to represent correct encryption schemes with IND-CPA security? (You do not need to explain how to decrypt.)

In the following, $\|$ denotes string concatenation and $\oplus$ bit-wise XOR.

(a) $\mathrm{Enc}_a(\mathsf{K}, m) = \mathrm{Enc}(\mathsf{K}, m) \,\|\, m[1]$.
   Here, $m[1]$ is the first bit of m.

(b) $\mathrm{Enc}_b(\mathsf{K}, m) = \mathrm{Enc}(\mathsf{K}, m \,\|\, r)$.
   Here, the message space for Enc_b is $\{0,1\}^{n/2}$, r is a per message random $n/2$-bit string, and $m \,\|\, r$ is the concatenation of m and r.

(c) $\mathrm{Enc}_c(\mathsf{K}, m) = \mathrm{Enc}(\mathsf{K}, m) \oplus \mathrm{Enc}(\mathsf{K}, 0^n)$.

(d) $\mathrm{Enc}_d(\mathsf{K}, m) = (\mathrm{Enc}(\mathsf{K}, m), \mathrm{Enc}(\mathsf{K}, m \oplus 1^n))$.
   That is, encrypt m, and then encrypt the bitwise complement of m.

(e) $\mathrm{Enc}_e(\mathsf{K}, m) = \{\mathrm{Enc}(\mathsf{K}, m), \mathrm{Enc}(\mathsf{K}, m \oplus 1^n)\}$.
   Here, $\{a, b\}$ denotes an unordered set containing elements a and b.

There are several cases:

(1) The new scheme is guaranteed to be secure, no matter how SE works (as long as it is IND-CPA secure). In this case, prove your claim.

(2) The new scheme is always insecure, no matter what SE does. In this case, show an attack that works no matter what.

(3) It may be the case that the new scheme is not a correct encryption scheme for some choices of SE. In this case explain why. Also explain: does the new encryption scheme leak any information about encrypted messages?

Problem 4 (PRFs). Let $k, l, L \in \mathbb{N}$ and $\mathsf{F}\colon \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^L$ be a pseudorandom function (PRF). For each of the new functions F_i defined below, determine if it is also a PRF. If you believe it is not, then provide an attack. Else justify that it is a PRF by an informal argument or, better, a proof by reduction. Note that some of the derived functions have domains or ranges that differ from that of F.

Let us also recall the definition of PRF security of a function F, in which an adversary $\mathcal{A}$ with access to an oracle $\textsc{Fn}$ is asked to distinguish a "real world" ($b = 0$) where $\textsc{Fn}(x)$ returns the evaluation $\mathsf{F}(\mathsf{K}, x)$ under a random key K from a "random world" ($b = 1$) where $\textsc{Fn}(x)$ returns the evaluation $f(x)$ under a randomly sampled function $f\colon \{0,1\}^l \to \{0,1\}^L$. The advantage of $\mathcal{A}$ in this game is defined as $\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{F}}(\mathcal{A}) = 2 \cdot \left|\Pr\left[\text{Game } \mathbf{PRF}(\mathcal{A}, \mathsf{F}) \Rightarrow \mathsf{true}\right] - \frac{1}{2}\right|$, where $\Pr\left[\text{Game } \mathbf{PRF}(\mathcal{A}, \mathsf{F}) \Rightarrow \mathsf{true}\right]$ denotes the probability that the output of Game **PRF** is true.

```
Game PRF(A, F):                         Oracle Fn(x):
1  $b \leftarrow_\$ \{0,1\}$                 6  If $b = 0$ then:
2  $\mathsf{K} \leftarrow_\$ \{0,1\}^k$          7      $y \leftarrow \mathsf{F}_\mathsf{K}(x)$
3  $f \leftarrow_\$ \mathrm{Funcs}[\{0,1\}^l, \{0,1\}^L]$   8  Else if $b = 1$ then:
4  $b' \leftarrow_\$ \mathcal{A}^{\textsc{Fn}}()$            9      $y \leftarrow f(x)$
5  Return $b' = b$                       10 Return $y$
```

In the following, $\|$ denotes string concatenation and $\oplus$ bit-wise XOR. For a bit-string $x \in \{0,1\}^\ell$, let $\overline{x}$ denote the bit-wise inversion and $|x| = \ell$ the bit-length of x.

(a) $\mathsf{F}_1(\mathsf{K}, x) := \mathsf{F}(\mathsf{K}, x) \,\|\, \overline{\mathsf{F}(\mathsf{K}, x)}$

(b) $\mathsf{F}_2(\mathsf{K}, x) := \mathsf{F}(\mathsf{K}, x) \,\|\, \mathsf{F}(\mathsf{K}, \overline{x})$

(c) $\mathsf{F}_3(\mathsf{K}, x) := \mathsf{F}(\mathsf{K}, x) \oplus 1^L$, where 1^L is the bit-string consisting of L 1s, for example $1^4 = 1111$.

(d) $\mathsf{F}_4(\mathsf{K}, x) := \mathsf{F}(\mathsf{K}, x) \oplus \mathsf{F}(\mathsf{K}, 1^l \oplus x)$

(e) $\mathsf{F}_5(\mathsf{K}_1 \,\|\, \mathsf{K}_2, x) := \mathsf{F}(\mathsf{K}_1, x) \oplus \mathsf{F}(\mathsf{K}_2, x)$, where F_5's key space is $\{0,1\}^{2k}$ and $|\mathsf{K}_1| = |\mathsf{K}_2| = k$

Acknowledgements. This exercise sheet is in part inspired by (and adapted from) problems by Mark Zhandry, as well as from the book "A Graduate Course in Applied Cryptography" by Dan Boneh and Victor Shoup, and from the book "The Joy of Cryptography" by Mike Rosulek.

References

[1] D. Boneh and V. Shoup. *A Graduate Course in Applied Cryptography.* Online, version 0.6 edition, Jan. 2023.