Introduction to Cryptography - Exercise session 3

Prof. Sebastian Faust

November 7, 2018

The purpose of this exercise session is to recall the concept of: a One-Way Function (OWF), a Pseudorandom Function (PRF) and a symmetric encryption scheme secure under the Chosen Plaintext Attack (CPA). For each of these primitives you can find the recap of the definition in a gray box.

ONE WAY FUNCTION

For a function $f\colon \{0,1\}^* \to \{0,1\}^*$ and for a ppt algorithm $\mathcal{A}$, define the inversion exepriment $\mathbf{Invert}_{\mathcal{A},f}(n)$ as follows:

$\mathbf{Invert}_{\mathcal{A},f}(n)$:

1. Choose $x \leftarrow \{0,1\}^n$ uniformly at random and compute $y := f(x)$.
2. $x' \leftarrow \mathcal{A}(1^n, y)$
3. If $f(x') = y$ output 1, else output 0.

Definition 1 (One Way Function) *A function $f\colon \{0,1\}^* \to \{0,1\}^*$ is one-way if the following holds*

1. *Easy to Compute: $\exists$ ppt algorithm $\mathcal{M}_f$, s.t. $\forall x \in \{0,1\}^*\colon \mathcal{M}_f(x) = f(x)$ and*
2. *Hard to Invert: $\forall$ ppt algorithms $\mathcal{A}$, $\exists$ negl s.t.*

$$\Pr[\mathbf{Invert}_{\mathcal{A},f} = 1] \leq \mathsf{negl}(n).$$

Solution:

Notation to be explained during the exercise session:

- $\mathbf{Invert}_{\mathcal{A},f}(n)$
  this is a probabilistic algorithm that is parametrized by a ppt algorithm $\mathcal{A}$ and a function f which on input n outputs a bit.
- $\Pr[\mathbf{Invert}_{\mathcal{A},f} = 1]$
  this denotes the probability that the experiment $\mathbf{Invert}_{\mathcal{A},f}(n)$ outputs 1. The probability is taken over the randomness of the algorithm $\mathbf{Invert}_{\mathcal{A},f}(n)$; more precisely, over the random choice of x (step 1) and the randomness of the ppt algorithm $\mathcal{A}$ (step 2).

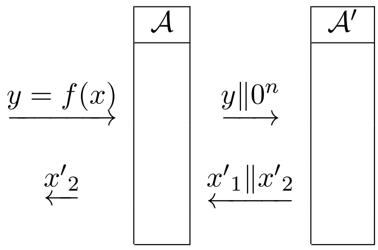
Exercise 1 (One-Way Functions)

Let f, g be arbitrary length-preserving one-way functions (i.e. $|f(x)| = |x|$). For each of the following functions f' decide, whether it is a OWF or not. If yes, give a proof else give a counter-example (assuming one-way functions exist, show that there are one-way function f, g such that f' is not a one-way function).

(a) $f'(x) = f(x) \oplus g(x)$.

> **Solution:**
>
> f' is not a OWF.
>
> We design a counter-example as follows. Fix $f(x) = g(x)$. This implies that $f'(x) = f(x) \oplus g(x) = 0$ for all x. Since f' is a constant function, is not a one-way function. (An adversary that outputs an arbitrary preimage x' always successfully wins the invert experiment.)

(b) $f'(x_1 \parallel x_2) = f(x_2) \parallel 0^n$.

> **Solution:**
>
> f' is a OWF. Proof by contradiction:
>
> For the sake of contradiction, let us assume that f' is not OWF. This implies that $\exists$ algorithm $\mathcal{A}'$ and a positive polynomial $p(n)$, s.t.
>
> $$\Pr[\mathbf{Invert}_{\mathcal{A}',f'}(n) = 1] > \frac{1}{p(n)}$$
>
> Now define an algorithm $\mathcal{A}$ as follows.
>
> 
>
>
> This implies that $\mathcal{A}$ is such that s.t.
>
> $$\Pr[\mathbf{Invert}_{\mathcal{A},f}(n) = 1] \geq \Pr[\mathbf{Invert}_{\mathcal{A}',f'}(n) = 1] > \frac{1}{p(n)}$$
>
> This is a contradiction with the assumption that f is a OWF.

(c) $f'(x) = f(f(x))$.

> **Solution:**
>
> $f'(x)$ is not a OWF.
>
> We design a counter-example as follows: Given a length preserving OWF g, by part (b) of this exercise, $f(x_1 \| x_2) := g(x_2) \| 0^n$ is a OWF. If f' is constructed using this function f, then we have
>
> $$f'(x_1 \| x_2) = f(f(x_1 \| x_2)) = f(g(x_2) \| 0^n) = g(0^n) \| 0^n,$$

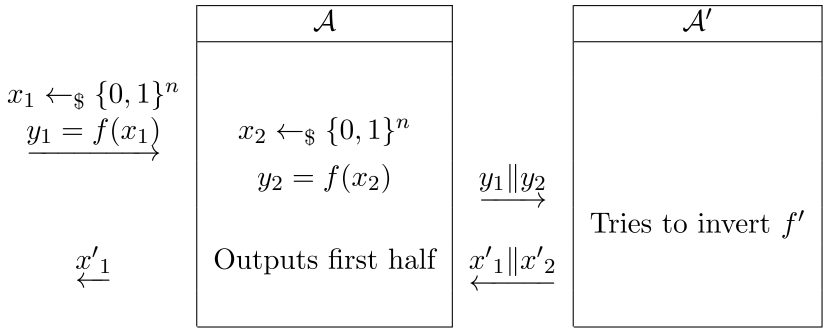
i.e. f' is a constant function and hence it is not a one-way function. (An adversary that outputs an arbitrary preimage $x'_1||x'_2$ always successfully wins the invert experiment.)

(d) $f'(x_1, x_2) = f(x_1) \parallel f(x_2)$.

Solution:
f' is a OWF. Direct proof:
For the function f', fix an algorithm $\mathcal{A}'$. Let us denote $\epsilon(n)$ as follows

$$\epsilon(n) := \Pr[\mathbf{Invert}_{\mathcal{A}',f'}(n) = 1]$$

Now construct an algorithm $\mathcal{A}$ in the following way



We can conclude from here that

$$\Pr[\mathbf{Invert}_{\mathcal{A},f} = 1] \geq \Pr[\mathbf{Invert}_{\mathcal{A}',f'} = 1] = \epsilon(n) \tag{1}$$

By definition f is OWF. It follows from equation (1) that f' is OWF.

PSEUDORANDOM FUNCTION

Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed function. F is a *pseudorandom function* if for all probabilistic polynomial-time distinguishers D, there exists a negligible function negl such that:

$$|\Pr[\mathsf{D}^{F_k(\cdot)}(1^n) = 1] - \Pr[\mathsf{D}^{f(\cdot)}(1^n) = 1]| \leq \mathsf{negl}(n)$$

where the first probability is taken over uniform choice of $k \in \{0,1\}^n$ and the randomness of D, and the second probability is taken over uniform choice of $f \in \mathsf{Func}_n$ and the randomness of D.

Solution:
Concepts to be explained during the exercise session:

- $\mathsf{D}^{\mathcal{O}(\cdot)}(1^n)$
  The distinguisher $\mathsf{D}^{\mathcal{O}(\cdot)}(1^n)$ is a ppt algorithm that gets as input the security parameter n. D has *oracle access* to a function $\mathcal{O}$. In other words, D can send a query

$x \in \{0,1\}^n$ to the oracle and receive $\mathcal{O}(x) \in \{0,1\}^n$ as an answer. The algorithm D can make polynomially many such queries. The output of the distinguisher is a bit.

- $\Pr[\mathsf{D}^{F_k(\cdot)}(1^n) = 1]$
  This denotes the probability that a distinguisher having an oracle access to the keyed function F_k outputs 1. The probability is taken over the random choice of the key k and the randomness of the distinguisher D.

- $\Pr[\mathsf{D}^{f(\cdot)}(1^n) = 1]$
  This denotes the probability that a distinguisher having an oracle access to a radnom function $f \in \mathsf{Func}_n$ outputs 1. The probability is taken over the random choice of the function f and the randomness of the distinguisher D.

- $\mathsf{Func}_n = \{f | f\colon \{0,1\}^n \to \{0,1\}^n\}$, i.e. Func_n is a set of all functions that take as input a bitstring of length n and output a bitstring of length n.

Exercise 2 (PRF)

For security parameter n, consider the following keyed function $F : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$. The key is a pair (k_1, k_2), where $k_1, k_2 \in \{0,1\}^n$ and F is defined by

$$F_{(k_1,k_2)}(x) := k_1 \oplus x \oplus k_2.$$

Show that F is not a PRF.

Solution:

We construct a distinguisher D as follows: On input 1^n and having access to oracle $\mathcal{O}$, D queries the oracle on 0^n and gets $c_0 := \mathcal{O}(0^n)$ as an answer and on 1^n and gets the answer $c_1 := \mathcal{O}(1^n)$. After that, D checks whether $c_0 \oplus c_1 = 1^n$ and if yes, then he outputs 1. Otherwise he outputs 0.

We will now prove that the constructed distinguisher D can with non-negligible probability distinguish between $\mathcal{O}$ being the keyed function F or a random function f.

If $\mathcal{O} = F_{(k_1,k_2)}$ for some (randomly chosen) k_1, k_2 , we have that

$$c_0 \oplus c_1 = \mathcal{O}(0^n) \oplus \mathcal{O}(1^n) = k_1 \oplus 0^n \oplus k_2 \oplus k_1 \oplus 1^n \oplus k_2 = 1^n$$

and therefore

$$\Pr_{(k_1,k_2) \leftarrow \{0,1\}^{2n}}[\mathsf{D}^{F_{(k_1,k_2)}(\cdot)}(1^n) = 1] = 1. \tag{2}$$

Now if $\mathcal{O}$ is a truly random function f, we have that $f(0^n)$ and $f(1^n)$ are random strings and hence $f(0^n) \oplus f(1^n)$ is also a random string. This implies that

$$\Pr_{f \leftarrow \mathsf{Func}(n)}[\mathsf{D}^{f(\cdot)}(1^n) = 1] = 2^{-n}.$$

We conclude that

$$|\Pr_{(k_1,k_2) \leftarrow \{0,1\}^{2n}}[\mathsf{D}^{F_{(k_1,k_2)}(\cdot)}(1^n) = 1] - \Pr_{f \leftarrow \mathsf{Func}(n)}[\mathsf{D}^{f(\cdot)}(1^n) = 1]| = 1 - 2^{-n}$$

which is clearly not negligible. It follows that F is not a PRF.

CPA-security

Consider the following experiment defined for any encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, adversary $\mathcal{A}$, and value n for the security parameter:

The CPA indistinguishability experiment $\mathbf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\mathbf{\Pi}}(\mathbf{n})$:

1. A key k is generated by running $\mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a pair of messages m_0, m_1 of the same length.
3. A uniform bit $b \in \{0,1\}$ is chosen, and then a ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$.
4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b_0 = b'$, and 0 otherwise. In the former case, we say that $\mathcal{A}$ succeeds.

Definition 2 (CPA security) *A private-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has indistinguishable encryptions under a chosen-plaintext attack, or is* CPA-secure*, if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there is a negligible function* negl *such that*

$$\Pr[\mathbf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

where the probability is taken over the randomness used by $\mathcal{A}$, as well as the randomness used in the experiment.

Solution:

To explain:

- $\mathbf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$
  this is a probabilistic algorithm that is parameterized by a ppt algorithm $\mathcal{A}$ and an encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The algorithm **PrivK** is denoted by a superscript cpa which indicates the notion of security considered. In this case it is *Chosen Plain-text Attack*, abbreviated as *cpa*. **PrivK** takes as input n and outputs a bit.
- $\Pr[\mathbf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1]$
  this denotes the probability that the experiment $\mathbf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$ outputs 1. The probability is taken over the randomness of the algorithm $\mathbf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$; more precisely, over the randomness of Gen, randomness of $\mathsf{Enc}_k(\cdot)$, random choice of the bit b, randomness of the algorithm $\mathcal{A}$.

Exercise 3 (CPA security - Combiner)

Let $\Pi_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ and $\Pi_2 = (\mathsf{Gen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$ be two encryption schemes for which it is known that at least one of them is CPA-secure (but you do not know which one). Show how to construct an encryption scheme Π that is guaranteed to be CPA-secure as long as at least one of Π_1, Π_2 is CPA-secure. Provide a full proof of your solution.

Solution:

Let n be a security parameter and let m be a message of length l. Let us define an encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such that

$$\begin{aligned} \mathsf{Gen}(1^n) &:= (\mathsf{Gen}_1(1^n), \mathsf{Gen}_2(1^n)) =: (k_1, k_2) =: k \\ \mathsf{Enc}(k; m) &:= (\mathsf{Enc}_1(k_1; s_1), \mathsf{Enc}_2(k_2; s_2)) =: (c_1, c_2) =: c \\ \mathsf{Dec}(k; c) &:= \mathsf{Dec}_1(k_1; c_1) \oplus \mathsf{Dec}_2(k_2; c_2) \end{aligned}$$

where s_1 is a a random string of length l and $s_2 := s_1 \oplus m$ (note that $s_2 \oplus s_1 = m$). We prove in the following that Π is CPA-secure.

Let us suppose by contradiction that Π is not CPA-secure. Then this means that there exists a PPT adversary $\mathcal{A}$ and a positive polynomial p such that

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] > \frac{1}{2} + 1/p(n) \tag{3}$$

where $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$ is the CPA indistinguishability experiment, defined in class. Let us denote $q(n)$ the number of encryption queries made by $\mathcal{A}$ before the challenge phase.

using adversary $\mathcal{A}$, we define a PPT adversary $\mathcal{A}_1$ for Π_1 as follows:

$\mathcal{A}$		$\mathcal{A}_1$		Challenger
		$k_2 \leftarrow \mathsf{Gen}_2(1^n)$		$k_1 \leftarrow \mathsf{Gen}_1(1^n)$
For $i \in [q(n)]$: $m^{(i)} \in \{0,1\}^n$	$\xrightarrow{m^{(i)}}$			
		$s^{(i)} \leftarrow_\$ \{0,1\}^n$ $x^{(i)} := s^{(i)} \oplus m^{(i)}$		
			$\xrightarrow{x^{(i)}}$	$y^{(i)} := \mathsf{Enc}_1(k_1; x^{(i)})$
			$\xleftarrow{y^{(i)}}$	
		$c^{(i)} :=$ $(y^{(i)}, \mathsf{Enc}_2(k_2; s^{(i)}))$		
	$\xleftarrow{c^{(i)}}$			
$m'_0, m'_1 \in \{0,1\}^n$	$\xrightarrow{m'_0, m'_1}$			
		$s_2 \leftarrow_\$ \{0,1\}^n$ $m_0 := s_2 \oplus m'_0$ $m_1 := s_2 \oplus m'_1$		
			$\xrightarrow{m_0, m_1}$	$b \leftarrow_\$ \{0,1\}$ $c_b := \mathsf{Enc}_1(k_1; m_b)$
			$\xleftarrow{c_b}$	
		$c'_b := (c_b, \mathsf{Enc}_2(k_2; s_2))$		
	$\xleftarrow{c'_b}$			
output : b'		output : b'		

Since $\mathcal{A}_1$ perfectly simulates the environment of a CPA-game for $\mathcal{A}$, we have that

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A}_1,\Pi_1}(n) = 1] = \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \overset{Eq.\ (3)}{>} \frac{1}{2} + 1/p(n). \tag{4}$$

Similarly, let us now define $\mathcal{A}_2$ a ppt adversary for Π_2 as in the following:

$\mathcal{A}$		$\mathcal{A}_2$		Challenger
		$k_1 \leftarrow \mathsf{Gen}_1(1^n)$		$k_2 \leftarrow \mathsf{Gen}_2(1^n)$
For $i \in [q(n)]$: $m^{(i)} \in \{0,1\}^n$	$\xrightarrow{m^{(i)}}$			
		$s^{(i)} \leftarrow_{\$} \{0,1\}^n$ $x^{(i)} := s^{(i)} \oplus m^{(i)}$		
			$\xrightarrow{x^{(i)}}$	$y^{(i)} := \mathsf{Enc}_2(k_2; x^{(i)})$
			$\xleftarrow{y^{(i)}}$	
		$c^{(i)} :=$ $(\mathsf{Enc}_1(k_1; s^{(i)}), y^{(i)})$		
	$\xleftarrow{c^{(i)}}$			
m'_0, m'_1	$\xrightarrow{m'_0, m'_1}$			
		$s_1 \leftarrow_{\$} \{0,1\}^n$ $m_0 := s_1 \oplus m'_0$ $m_1 := s_1 \oplus m'_1$		
			$\xrightarrow{m_0, m_1}$	$b \leftarrow_{\$} \{0,1\}$ $c_b := \mathsf{Enc}_2(k_2; m_b)$
			$\xleftarrow{c_b}$	
		$c'_b := (\mathsf{Enc}_1(k_1; s_1), c_b)$		
	$\xleftarrow{c'_b}$			
output : b'		output : b'		

Since $\mathcal{A}_2$ perfectly simulates the environment of a CPA-game for $\mathcal{A}$, we have that

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A}_2,\Pi_2}(n) = 1] = \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \overset{Eq.\ (3)}{>} \frac{1}{2} + 1/p(n). \tag{5}$$

We proved that if an adversary $\mathcal{A}$ exists, then neither of the schemes Π_1 and Π_2 is CPA-secure, which contradicts our the hypothesis.

Exercise 4 (CPA-security - Voluntary homework exercise)

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a deterministic, stateless symmetric encryption scheme. Then the scheme Π is not CPA-secure.

Solution:

Let us construct an adversary $\mathcal{A}$, such that $\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] > \frac{1}{2} + \mathsf{negl}(n)$.

$\mathcal{A}$		*Oracle*
		$k \leftarrow \mathsf{Gen}(1^n)$
$m \in \{0,1\}^n$	$\xrightarrow{m}$	$c \leftarrow \mathsf{Enc}(k; m)$
	$\xleftarrow{c}$	
$m_1 := m$ $m_0 \in \{0,1\}^n$ s.t. $m_0 \neq m$	$\xrightarrow{m_0, m_1}$	$b \leftarrow_{\$} \{0,1\}$ $c_b \leftarrow \mathsf{Enc}(k; m_b)$
	$\xleftarrow{c_b}$	
If $c_b = c$, $\mathsf{output}: 1$ Else $\mathsf{output}: 0$		

If $c_b = \mathsf{Enc}(k, m_1)$, then, since the encryption function is deterministic and stateless, $c = c_b'$. Therefore $\mathcal{A}$ always correctly outputs 1 in this case. If $c_b = \mathsf{Enc}(k, m_0)$, then $\mathcal{A}$ always correctly outputs 0. This is because $m_1 \neq m_0$ which implies $c_b \neq c$ (correctness implies that encryption of two different messages must result in two different ciphertexts)

Overall we get

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] = 1 - 0 = 1 > \frac{1}{2} + \mathsf{negl}(n)$$

completing the proof.