TD 1 Crytography Engineering

Léo Colisson Palais

Exercice 1:

1. Compute $\Pr[\mathcal{A}_1 \diamond \mathcal{L}_1 = \textsf{true}]$, $\Pr[\mathcal{A}_1 \diamond \mathcal{L}_2 = \textsf{true}]$, $\Pr[\mathcal{A}_2 \diamond \mathcal{L}_1 = \textsf{true}]$, $\Pr[\mathcal{A}_2 \diamond \mathcal{L}_2 = \textsf{true}]$ with

$\mathcal{A}_1$
$r_1 \leftarrow \text{RAND}(6)$
$r_2 \leftarrow \text{RAND}(6)$
return $r_1 \stackrel{?}{=} r_2$

$\mathcal{A}_2$
$r_1 \leftarrow \text{RAND}(6)$
$r_2 \leftarrow \text{RAND}(6)$
return $r_1 \stackrel{?}{\geq} 3$

$\mathcal{L}_1$
$\text{RAND}(n)$:
$\quad r \xleftarrow{\$} \mathbb{Z}_n$
$\quad$ return r

$\mathcal{L}_2$
$\text{RAND}(n)$:
$\quad$ return 0

2. Are the following libraries indistinguishable? (if so, describe the distinguisher (you can test it in Caseine) and **compute** its success probability:

(a) $\mathcal{L}_A \stackrel{?}{\approx} \mathcal{L}_B$

$\mathcal{L}_A$	$\mathcal{L}_B$
$\text{SAMPLE}()$:	$\text{SAMPLE}()$:
$\quad$ **return** 42	$\quad$ **return** 45

(b) $\mathcal{L}_A \stackrel{?}{\approx} \mathcal{L}_B$

$\mathcal{L}_A$	$\mathcal{L}_B$
$\text{SAMPLE}()$:	$\text{SAMPLE}()$:
$\quad x \xleftarrow{\$} \mathbb{Z}_{10}$	$\quad x \xleftarrow{\$} \mathbb{Z}_9$
$\quad$ **return** x	$\quad$ **return** x

(c) $\mathcal{L}_A \stackrel{?}{\approx} \mathcal{L}_B$

$\mathcal{L}_A$	$\mathcal{L}_B$
$\text{SAMPLE}()$:	$\text{SAMPLE}()$:
$\quad x \xleftarrow{\$} \mathbb{Z}_{10}$	$\quad x \xleftarrow{\$} \mathbb{Z}_{10}$
$\quad$ **return** x	$\quad$ **return** $2 + x$

(d) $\mathcal{L}_A \stackrel{?}{\approx} \mathcal{L}_B$

$\mathcal{L}_A$	$\mathcal{L}_B$
$\text{SAMPLE}()$:	$\text{SAMPLE}()$:
$\quad x \xleftarrow{\$} \mathbb{Z}_{10}$	$\quad x \xleftarrow{\$} \mathbb{Z}_{10}$
$\quad$ **return** x	$\quad$ **return** $(2 + x) \mod 10$

(e) $\mathcal{L}_A \stackrel{?}{\approx} \mathcal{L}_B$

$\mathcal{L}_A$	$\mathcal{L}_B$
$a := 0$	$a := 1$
$\text{SAMPLE}()$:	$\text{SAMPLE}()$:
$\quad$ **return** 42	$\quad b := 8$
	$\quad$ **return** 42

(f) $\mathcal{L}_A \stackrel{?}{\approx} \mathcal{L}_B \diamond \mathcal{L}_C$

$\mathcal{L}_A$	$\mathcal{L}_B$	$\mathcal{L}_C$
$\text{SAMPLE}()$:	$\text{SAMPLE}()$:	$\text{SQUARE}(x)$:
$\quad$ **return** 9	$\quad$ **return** $\text{SQUARE}(3)$	$\quad$ **return** x^2

(g) $\mathcal{L}_A \stackrel{?}{\approx} \mathcal{L}_B$

$\mathcal{L}_A$	$\mathcal{L}_B$
$\text{SAMPLE}()$:	$\text{SAMPLE}()$:
$\quad x \xleftarrow{\$} \mathbb{Z}_{10}$	$\quad x \xleftarrow{\$} \mathbb{Z}_{10}$
$\quad$ **if** $x \stackrel{?}{=} 0$ **then** $x \xleftarrow{\$} \mathbb{Z}_{10}$	$\quad$ **return** x
$\quad$ **return** x	

(h) The libraries of the IND-CPA security definition with the encryption scheme $\textsf{Gen}(1^\lambda)$ always returning 0, and $\textsf{Enc}_k(m) := \bar{m}$, where $m \in \{0, 1\}^\lambda$, $\bar{m}$ is the bitwise flip of m (0 becomes 1 and 1 becomes 0).

(i) The libraries of the IND-CPA security definition with the One-Time Pad encryption scheme.

(j) The libraries of the IND-CPA security definition with any unknown deterministic encryption scheme.

Exercice 2: Negligible functions and library manipulation

1. Which of the following functions are negligible? (justify, you may use $a^b = 2^{b \log a}$)

$$\frac{1}{2^{\lambda/2}} \qquad \frac{1}{2^{\log(\lambda^2)}} \qquad \frac{1}{\lambda^{\log(\lambda)}} \qquad \frac{1}{\lambda^2} \qquad \frac{1}{2^{\log \lambda^2}} \qquad \frac{1}{\lambda^{1/\lambda}} \qquad \frac{1}{\sqrt{\lambda}} \qquad \frac{1}{2^{\sqrt{\lambda}}}$$

2. Show that if f and g are negligible, so are $f + g$ and fg.

3. Show that if $f = \mathsf{poly}(\lambda)$ and $g = \mathsf{negl}(\lambda)$, $fg = \mathsf{negl}(\lambda)$.

Exercice 3: A simple secret sharing scheme

We consider the following libraries:

$$\begin{array}{|l|}\hline \mathcal{L}_{\text{ot-real}} \\ \hline \underline{\text{QUERY}(m \in \{0,1\}^\lambda):} \\ \quad r \xleftarrow{\$} \{0,1\}^\lambda \\ \quad y := r \oplus m \\ \quad \text{return } y \\ \hline \end{array} \qquad \begin{array}{|l|}\hline \mathcal{L}_{\text{ot-rand}} \\ \hline \underline{\text{QUERY}(m \in \{0,1\}^\lambda):} \\ \quad r \xleftarrow{\$} \{0,1\}^\lambda \\ \quad \text{return } r \\ \hline \end{array} \qquad \begin{array}{|l|}\hline \mathcal{L}_{\text{left}} \\ \hline \underline{\text{QUERY}(m \in \{0,1\}^\lambda):} \\ \quad r \xleftarrow{\$} \{0,1\}^\lambda \\ \quad y := r \oplus m \\ \quad \text{return } (r, y) \\ \hline \end{array} \qquad \begin{array}{|l|}\hline \mathcal{L}_{\text{right}} \\ \hline \underline{\text{QUERY}(m \in \{0,1\}^\lambda):} \\ \quad r \xleftarrow{\$} \{0,1\}^\lambda \\ \quad y := r \oplus m \\ \quad \text{return } (y, r) \\ \hline \end{array}$$

1. Show that $\mathcal{L}_{\text{ot-real}} \equiv \mathcal{L}_{\text{ot-rand}}$ Hint: use the "compute the probability" method.

2. Use it to give different proof that the one-time pad (OTP) is one-time secure.

$$\begin{array}{|l|}\hline \mathcal{L}^{\Sigma}_{\text{ots-L}} \\ \hline \underline{\text{EAVESDROP}(m_L, m_R \in \mathcal{M}):} \\ \quad k \leftarrow \mathsf{Gen}(1^\lambda) \\ \quad \text{return } \mathsf{Enc}_k(m_L) \\ \hline \end{array} \equiv \begin{array}{|l|}\hline \mathcal{L}^{\Sigma}_{\text{ots-R}} \\ \hline \underline{\text{EAVESDROP}(m_L, m_R \in \mathcal{M}):} \\ \quad k \leftarrow \mathsf{Gen}(1^\lambda) \\ \quad \text{return } \mathsf{Enc}_k(m_R) \\ \hline \end{array}$$

3. Show that $\mathcal{L}_{\text{left}} \equiv \mathcal{L}_{\text{right}}$. Can you use directly the fact that $\mathcal{L}_{\text{ot-real}} \equiv \mathcal{L}_{\text{ot-rand}}$? If yes, prove it, otherwise, show where the naive proof fails.

4. A t-out-of-n threshold secret-sharing scheme (TSSS) consists of two algorithms

   - $\mathsf{Share}(m \in \mathcal{M})$ that outputs a sequence $s = (s_1, \ldots, s_n)$ of shares,
   - $\mathsf{Reconstruct}(\{s_1, \ldots, s_k\})$ that outputs a message $m \in \mathcal{M}$ if $k \geq t$ and $\perp$ otherwise.

   such that:

   - Correctness: for any $m \in \mathcal{M}$ and $U \subseteq \{1, \ldots, n\}$ such that $|U| \geq t$, and for all $s \leftarrow \mathsf{Share}(m)$, we have $\mathsf{Reconstruct}(\{s_i \mid i \in U\}) = m$,
   - Security: we have

$$\begin{array}{|l|}\hline \mathcal{L}_{\text{tsss-L}} \\ \hline \underline{\text{SHARE}(m_L, m_R, U):} \\ \quad \text{if } |U| \geq t, \text{ return } \texttt{err} \\ \quad s \leftarrow \text{SHARE}(m_L) \\ \quad \text{return } \{s_i \mid i \in U\} \\ \hline \end{array} \equiv \begin{array}{|l|}\hline \mathcal{L}_{\text{tsss-R}} \\ \hline \underline{\text{SHARE}(m_L, m_R, U):} \\ \quad \text{if } |U| \geq t, \text{ return } \texttt{err} \\ \quad s \leftarrow \text{SHARE}(m_R) \\ \quad \text{return } \{s_i \mid i \in U\} \\ \hline \end{array} \tag{1}$$

   (a) Explain why this is called a "secret-sharing scheme".

(b) Is the following construction secure? If yes, proves it, otherwise, find an explicit attacker.

$\mathcal{M} = \{0,1\}^{500}$ $t = 5$ $n = 5$	$\underline{\mathsf{Share}(m)}$: split m into $m = s_1 \| \cdots \| s_5$, where each $\lvert s_i \rvert = 100$ return $(s_1, \dots, s_5)$	$\underline{\mathsf{Reconstruct}(s_1, \dots, s_5)}$: return $s_1 \| \cdots \| s_5$

(c) We consider a simple 2-out-of-2 secret sharing scheme, where Share is defined as the QUERY in $\mathcal{L}_{\text{left}}$. Describe the Reconstruct procedure.

(d) Prove that this scheme is secure.

Hint: do a case distinction on the size/value of U

(e) Can you generalize this construction to obtain a 2-out-of-k secret sharing scheme for arbitrary $k \in \mathbb{N}^*$ and prove its security?

Exercice 4: Security of OTP

1. Someone realizes that the OTP leaks the message when the key is $0 \dots 0$, and proposes to sample the key on $\{0,1\}^\lambda \setminus \{0^\lambda\}$ instead of $\{0,1\}^\lambda$. Is this more (or less?) secure? If yes, prove it, otherwise find an attacker attacking the one-time security of the scheme (i.e. the adversary should distinguish $\mathcal{L}^{\Sigma}_{\text{ots-L}}$ from $\mathcal{L}^{\Sigma}_{\text{ots-R}}$).

2. To get additional security, Alice decides to encrypt the message twice with OTP. What are the actual impacts in term of security *(i)* if Alice uses the same k for both encryptions *(ii)* if Alice uses different keys?

3. What is so special regarding the OTP's XOR function? Would it be correct and/or secure with, say, a *AND* instead of a XOR? Would it work if we interpret strings as integers modulo 2^λ and replace the XOR with a modular addition? (prove formally any statements)

4. Show that the following encryption scheme does not have one-time secrecy, by constructing a program that distinguishes the two relevant libraries from the one-time secrecy definition.

$\mathcal{K} = \{1, \dots, 9\}$ $\mathcal{M} = \{1, \dots, 9\}$ $\mathcal{C} = \mathbb{Z}_{10}$	$\underline{\mathsf{Gen}}$: $k \leftarrow \{1, \dots, 9\}$ return k	$\underline{\mathsf{Enc}(k, m)}$: return $k \times m \% 10$

5. You (Eve) have intercepted two ciphertexts:

$$c_1 = \texttt{1111100101111001110011000001011110000110}$$
$$c_2 = \texttt{1111101001100111110111010000100110001000}$$

You know that both are OTP ciphertexts, encrypted with the *same* key. You know that either *(i)* c_1 is an encryption of `alpha` and c_2 is an encryption of `bravo` or *(ii)* c_1 is an encryption of `delta` and c_2 is an encryption of `gamma` (all converted to binary from ascii in the standard way, i.e. $\texttt{a} = 97, \texttt{b} = 98 \dots$). Which of these two possibilities is correct, and why? Can you recover the key?