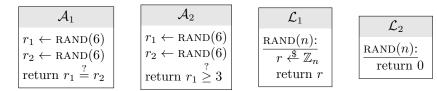
# TD 1 Crytography Engineering

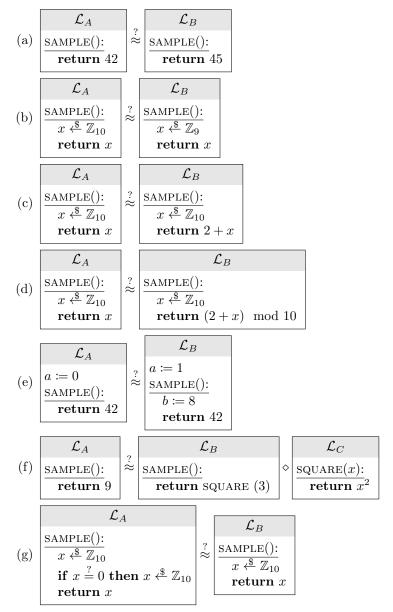
### Léo Colisson Palais

#### Exercice 1:

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



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



(h) The libraries of the IND-CPA security definition with the encryption scheme  $\mathsf{Gen}(1^{\lambda})$  always returning 0, and  $\mathsf{Enc}_k(m) \coloneqq \bar{m}$ , 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

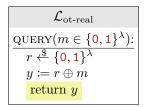
1. Which of the following functions are negligible? (justify, you may use  $a^b = 2^{b \log a}$ )

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

- 2. Show that if f and g are negligible, so are f + g and fg.
- 3. Show that if  $f = \mathsf{poly}(\lambda)$  and  $g = \mathsf{negl}(\lambda)$ ,  $fg = \mathsf{negl}(\lambda)$ .

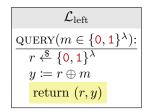
## Exercice 3: A simple secret sharing scheme

We consider the following libraries:



$$\frac{\mathcal{L}_{\text{ot-rand}}}{\frac{\text{QUERY}(m \in \{0, 1\}^{\lambda}):}{r \overset{\$}{\leftarrow} \{0, 1\}^{\lambda}}}$$

$$\frac{r \text{eturn } r}{r}$$



$$egin{aligned} \mathcal{L}_{ ext{right}} \ & rac{ ext{QUERY}(m \in \{0,1\}^{\lambda}):}{r \overset{\$}{\leftarrow} \{0,1\}^{\lambda}} \ y \coloneqq r \oplus m \ & ext{return } (y,r) \end{aligned}$$

- 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.

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

- 3. Show that  $\mathcal{L}_{left} \equiv \mathcal{L}_{right}$ . Can you use directly the fact that  $\mathcal{L}_{ot\text{-real}} \equiv \mathcal{L}_{ot\text{-rand}}$ ? 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
  - Share $(m \in \mathcal{M})$  that outputs a sequence  $s = (s_1, \ldots, s_n)$  of shares,
  - 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  $U \subseteq \{1, ..., n\}$  such that  $|U| \ge t$ , and for all  $s \leftarrow \mathsf{Share}(m)$ , we have  $\mathsf{Reconstruct}(\{s_i \mid i \in U\}) = m$ ,
- Security: we have

$$\frac{\mathcal{L}_{tsss-L}}{\underset{\text{if } |U| \geq t, \text{ return err}}{\text{err}}} = \frac{\mathcal{L}_{tsss-R}}{\underset{\text{share}(m_L, m_R, U):}{\text{sif } |U| \geq t, \text{ return err}}} = \frac{\mathcal{L}_{tsss-R}}{\underset{\text{share}(m_L, m_R, U):}{\text{if } |U| \geq t, \text{ return err}}} = (1)$$

$$\frac{\text{Share}(m_L, m_R, U):}{\underset{\text{share}(m_L)}{\text{share}(m_R)}} = \frac{\mathcal{L}_{tsss-R}}{\underset{\text{share}(m_L, m_R, U):}{\text{share}(m_L, m_R, U):}} = \frac{\mathcal{L}_{tsss-R}}{\underset{\text{share}(m_L, m_R, U):}{\text{share}(m_R, M_R, U):}} = \frac{\mathcal{L}_{tsss-R}}{\underset{\text{share}(m_R, M_R, U):}{\text{share}(m_R, M_R, U):}} = \frac{\mathcal{L$$

(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} \qquad \frac{\text{Share}(m):}{\text{split } m \text{ into } m = s_1 \| \cdots \| s_5,} \\ t = 5 \qquad \text{where each } |s_i| = 100 \qquad \frac{\text{Reconstruct}(s_1, \dots, s_5):}{\text{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  $\mathcal{L}_{\mathrm{left}}$ . Describe the Reconstruct procedure.
- (d) Prove that this scheme is secure.

  O to anistylation of the first of the secure of
- (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...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  $\mathcal{L}_{\text{ots-L}}^{\Sigma}$  from  $\mathcal{L}_{\text{ots-R}}^{\Sigma}$ ).
- 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  $2^{\lambda}$  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.

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

5. You (Eve) have intercepted two ciphertexts:

```
c_1 = 1111100101111001110011000001011110000110

c_2 = 11111010011001111110111010000100110001000
```

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. a = 97, b = 98...). Which of these two possibilities is correct, and why? Can you recover the key?