TD 1 Cryptography Engineering 2024

Léo Colisson Palais

Exercice 1: Negligible functions and library manipulation

1. Which of the following functions are negligible? Sort them from the smallest to the largest (asymptotically). Justify your answers.

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

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

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

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

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

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

$\mathcal{L}_1$
$\textsc{rand}(n)$:
$r \xleftarrow{\$} \mathbb{Z}_n$
return r

$\mathcal{L}_2$
$\textsc{rand}(n)$:
return 0

Exercice 2: A simple secret sharing scheme

We consider below the following libraries:

$\mathcal{L}_{\text{ot-real}}$
$\textsc{query}(m \in \{0,1\}^\lambda)$:
$r \xleftarrow{\$} \{0,1\}^\lambda$
$y := r \oplus m$
return y

$\mathcal{L}_{\text{ot-rand}}$
$\textsc{query}(m \in \{0,1\}^\lambda)$:
$r \xleftarrow{\$} \{0,1\}^\lambda$
return r

$\mathcal{L}_{\text{left}}$
$\textsc{query}(m \in \{0,1\}^\lambda)$:
$r \xleftarrow{\$} \{0,1\}^\lambda$
$y := r \oplus m$
return (r, y)

$\mathcal{L}_{\text{right}}$
$\textsc{query}(m \in \{0,1\}^\lambda)$:
$r \xleftarrow{\$} \{0,1\}^\lambda$
$y := r \oplus m$
return (y, r)

1. Show that $\mathcal{L}_{\text{ot-real}} \equiv \mathcal{L}_{\text{ot-rand}}$. Use it to give different proof that the one-time pad (OTP) is one-time secure.

2. 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.

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

   - $\mathsf{Share}(m \in \mathcal{M})$ that outputs a sequence $s = (s_1, \dots, s_n)$ of shares,
   - $\mathsf{Reconstruct}(\{s_1, \dots, s_k\})$ 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 $U \subseteq \{1, \dots, n\}$ 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

$\mathcal{L}_{\text{tsss-L}}$
$\textsc{share}(m_L, m_R, U)$:
if $\lvert U\rvert \geq t$, return err
$s \leftarrow \textsc{share}(m_L)$
return $\{s_i \mid i \in U\}$

$\equiv$

$\mathcal{L}_{\text{tsss-R}}$
$\textsc{share}(m_L, m_R, U)$:
if $\lvert U\rvert \geq t$, return err
$s \leftarrow \textsc{share}(m_R)$
return $\{s_i \mid i \in U\}$

(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 $(s_1, \ldots, s_5)$	$\underline{\mathsf{Reconstruct}(s_1, \ldots, s_5)}$: 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 3: Security of OTP

1. Someone realizes that the OTP leaks the message when the key is $0 \ldots 0$, 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.

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

5. You (Eve) have intercepted two ciphertexts:

$$c_1 = \texttt{11111001011111001110011000001011110000110}$$
$$c_2 = \texttt{11111010011001111110111010000100110001000}$$

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. $\mathtt{a} = 97, \mathtt{b} = 98 \ldots$). Which of these two possibilities is correct, and why? Can you recover the key?

Exercice 4: PRG extension and application to ratchet

We want to build a larger PRG H from a smaller length-doubling PRG $G\colon \{0,1\}^\lambda \to \{0,1\}^\lambda \times \{0,1\}^\lambda$. Here are 3 candidates:

$\underline{H_0(s)}$: $m \leftarrow G(s)$ return $m \| m$	$\underline{H_1(s)}$: $x \| y \leftarrow G(s)$ $u \| v \leftarrow G(s)$ return $x \| u \| v$	$\underline{H_2(s)}$: $x \| y \leftarrow G(s)$ $u \| v \leftarrow G(s)$ return $x \| y \| u \| v$

1. Which candidate is insecure (find an attack) and secure (prove it)? Why can't you apply the same proof for the other candidates?

2. Can you generalize the construction to arbitrarily large (polynomial) length extension?

★ 3. Describe (and prove) how this can be used to build a ratchet, i.e. an encryption mechanism that can even protect messages sent before a complete corruption of a party (leaking also the key).