Exam Cryptography Engineering 2024

Léo Colisson Palais

Exercice 1: Combining encryptions for better security

Alice and Bob want to securely exchange a message $m \in \mathcal{M} := \{0,1\}^*$. They have access to two encryption schemes $(\mathsf{Gen}_0 \colon \mathbb{N} \to \mathcal{K}_0, \mathsf{Enc}_0 \colon \mathcal{K}_0 \times \mathcal{M} \to \mathcal{C}_0, \mathsf{Dec}_0 \colon \mathcal{K}_0 \times \mathcal{C}_0 \to \mathcal{M})$ and $(\mathsf{Gen}_1 \colon \mathbb{N} \to \mathcal{K}_1, \mathsf{Enc}_1 \colon \mathcal{K}_1 \times \mathcal{M} \to \mathcal{C}_1, \mathsf{Dec}_1 \colon \mathcal{K}_1 \times \mathcal{C}_1 \to \mathcal{M})$, but they only know that at least one of them is secure, without knowing which one is actually secure. In the following exercise, we will study how to perform this securely.

1. (0.5 pts) Here are 3 equivalences between libraries[1]: which one corresponds to the security definition of indistinguishability under (variable-length plaintext) chosen plaintext attack (IND-CPA)? In the following, we will name the corresponding libraries as, respectively, $\mathcal{L}^{\mathsf{Gen},\mathsf{Enc}}_{\text{cpa-L}}$ and $\mathcal{L}^{\mathsf{Gen},\mathsf{Enc}}_{\text{cpa-R}}$.

③ EAVESDROP$(m_L, m_R \in \mathcal{M})$: ④ if $\lvert m_L\rvert \neq \lvert m_R\rvert$ return err ⓪ $k \leftarrow \mathsf{Gen}(1^\lambda)$ 5L $c := \mathsf{Enc}(k, m_L)$ ⑦ return c	$\approx$	③ EAVESDROP$(m_L, m_R \in \mathcal{M})$: ④ if $\lvert m_L\rvert \neq \lvert m_R\rvert$ return err ⓪ $k \leftarrow \mathsf{Gen}(1^\lambda)$ 5R $c := \mathsf{Enc}(k, m_R)$ ⑦ return c

(1)

① $k \leftarrow \mathsf{Gen}(1^\lambda)$ ③ EAVESDROP$(m_L, m_R \in \mathcal{M})$: ④ if $\lvert m_L\rvert \neq \lvert m_R\rvert$ return err 5L $c := \mathsf{Enc}(k, m_L)$ ⑦ return c	$\approx$	① $k \leftarrow \mathsf{Gen}(1^\lambda)$ ③ EAVESDROP$(m_L, m_R \in \mathcal{M})$: ④ if $\lvert m_L\rvert \neq \lvert m_R\rvert$ return err 5R $c := \mathsf{Enc}(k, m_R)$ ⑦ return c

(2)

① $k \leftarrow \mathsf{Gen}(1^\lambda)$ ② $\mathcal{S} := \emptyset$ ③ EAVESDROP$(m_L, m_R \in \mathcal{M})$: ④ if $\lvert m_L\rvert \neq \lvert m_R\rvert$ return err 5L $c := \mathsf{Enc}(k, m_L)$ ⑥ $\mathcal{S} := \mathcal{S} \cup \{c\}$ ⑦ return c ⑧ DECRYPT$(c \in \mathcal{C})$: ⑨ if $c \in \mathcal{S}$ return err ⑩ return $\mathsf{Dec}(k, c)$	$\approx$	① $k \leftarrow \mathsf{Gen}(1^\lambda)$ ② $\mathcal{S} := \emptyset$ ③ EAVESDROP$(m_L, m_R \in \mathcal{M})$: ④ if $\lvert m_L\rvert \neq \lvert m_R\rvert$ return err 5R $c := \mathsf{Enc}(k, m_R)$ ⑥ $\mathcal{S} := \mathcal{S} \cup \{c\}$ ⑦ return c ⑧ DECRYPT$(c \in \mathcal{C})$: ⑨ if $c \in \mathcal{S}$ return err ⑩ return $\mathsf{Dec}(k, c)$

(3)

Correction. The definition corresponding to CPA security is the second definition (eq. (2)). □

2. (0.5 pts) For each of these three security definitions, specify if the One-Time Pad (OTP) encryption scheme is secure according to this definition (we temporarily assume for simplicity that the message space $\mathcal{M}$ is equal to the key space $\mathcal{K}$ and $\mathcal{M} = \mathcal{K} = \{0,1\}^n$). No formal proof is expected here, but justify your answer with one or two sentences.

Correction. (a) OTP is secure according to the first definition, as this corresponds to the notion of one-time secrecy seen in the course: OTP is known to be secure if a fresh key is picked at every new encryption.

(b) OTP is NOT secure according to the second definition (CPA security): the same key is reused multiple times, and OTP is know to be unsecure if the key is reused more than once.

[1] The gray numbers like ① are just used to label lines so that you can quickly refer to them without rewriting them fully.

(c) OTP is NOT secure according to the last definition (CCA security), since the adversary can apply the same attack as in the CPA security definition.

□

3. (1.25 pts) To securely encrypt a message using Enc_0 and Enc_1 without knowing which one is actually secure, Alice proposes to encrypt m as follows:

$$\mathsf{Enc}(k := (k_0, k_1), m) := \mathsf{Enc}_1(k_1, \mathsf{Enc}_0(k_0, m)) \tag{4}$$

where keys (k_0, k_1) are generated by a procedure $(k_0, k_1) \leftarrow \mathsf{Gen}(1^\lambda)$ by sampling $k_0 \leftarrow \mathsf{Gen}_0(1^\lambda)$ and $k_1 \leftarrow \mathsf{Gen}_1(1^\lambda)$.

Assuming that Enc_1 is IND-CPA secure and that $|\mathsf{Enc}_1(k, m)| = l|m|$ for some integer $l \geq 1$, formally show that there exists Enc_0 such that Enc is *not* IND-CPA secure (more precisely, exhibit a function Enc_0 and an adversary $\mathcal{A}$ following the Joy of Cryptography notation seen in the course, and compute its advantage according to the IND-CPA security definition).

Hint: you can choose Enc_0 arbitrarily, in particular it may not preserve the length of its input.

Correction. We define $\mathcal{K}_0 = \{0\}$ (a single key $k_0 = 0$ is defined as we won't use it), $\mathsf{Gen}(1^\lambda)$ returns $k_0 := 0$, and $\mathsf{Enc}_0(k, m) := m$ if m starts with a 0 and $\mathsf{Enc}_0(k, m) := m \| m$ otherwise. Hence $|\mathsf{Enc}_0(k, 0)| = 1$ and $|\mathsf{Enc}_0(k, 1)| = 2$. We also define

$$\boxed{\begin{array}{c}\mathcal{A} \\ \hline \textbf{return } |\textsc{eavesdrop}(0,1)| \stackrel{?}{=} 2l\end{array}} \tag{5}$$

We compute now the advantage of $\mathcal{A}$. We can simplify

$$\mathcal{A} \diamond \mathcal{L}_{\text{cpa-L}} \equiv \boxed{\begin{array}{l} k_0 := 0 \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \textbf{return } |\mathsf{Enc}_1(k_0, \mathsf{Enc}_0(k_1, 0))| \stackrel{?}{=} 2l \end{array}} \tag{6}$$

But since $|\mathsf{Enc}_1(k_0, \mathsf{Enc}_0(k_1, 0))| = l|\mathsf{Enc}_0(k_1, 0)| = l$, we have $\Pr\left[\mathcal{A} \diamond \mathcal{L}_{\text{cpa-L}} \stackrel{?}{=} \mathsf{true}\right] = 0$. Similarly:

$$\mathcal{A} \diamond \mathcal{L}_{\text{cpa-R}} \equiv \boxed{\begin{array}{l} k_0 := 0 \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \textbf{return } |\mathsf{Enc}_1(k_0, \mathsf{Enc}_0(k_1, 1))| \stackrel{?}{=} 2l \end{array}} \tag{7}$$

but since $|\mathsf{Enc}_1(k_0, \mathsf{Enc}_0(k_1, 1))| = l|\mathsf{Enc}_0(k_1, 1)| = 2l$, we have $\Pr\left[\mathcal{A} \diamond \mathcal{L}_{\text{cpa-R}} \stackrel{?}{=} \mathsf{true}\right] = 1$. Hence, the advantage of $\mathcal{A}$ is

$$\left|\Pr\left[\mathcal{A} \diamond \mathcal{L}_{\text{cpa-R}} \stackrel{?}{=} \mathsf{true}\right] - \Pr\left[\mathcal{A} \diamond \mathcal{L}_{\text{cpa-L}} \stackrel{?}{=} \mathsf{true}\right]\right| = |1 - 0| = 1 \tag{8}$$

which is non-negligible. Hence, Enc is not IND-CPA secure, concluding the proof. □

4. In order to avoid the above attack, Alice has the idea to use a so-called secret-sharing operation to split the message m into two "shares" m_0 and m_1 such that $m = m_0 \oplus m_1$, and encrypt m_0 with Enc_0 and m_1 with Enc_1. More precisely, we consider the procedure $(k_0, k_1) \leftarrow \mathsf{Gen}(1^\lambda)$ defined above and the encryption as:

$$\boxed{\begin{array}{l}\mathsf{Enc}(k := (k_0, k_1), m) \\ \hline (11)\ \ m_0 \stackrel{\$}{\leftarrow} \{0,1\}^{|m|} \\ (12)\ \ m_1 := m_0 \oplus m \\ (13)\ \ \textbf{return } (\mathsf{Enc}_0(k_0, m_0), \mathsf{Enc}_1(k_1, m_1))\end{array}} \tag{9}$$

(a) (0.75 pts) Describe the decryption algorithm Dec and prove its correctness, i.e. that:

$$\Pr\left[\mathsf{Dec}((k_0, k_1), \mathsf{Enc}((k_0, k_1), m)) = m\right] = 1 \tag{10}$$

Correction. We define

$$\boxed{\begin{array}{l}\mathsf{Dec}(k := (k_0, k_1), (c_0, c_1)) \\ \textbf{return } \mathsf{Dec}_0(k_0, c_0) \oplus \mathsf{Dec}_1(k_1, c_1)\end{array}} \tag{11}$$

This decryption is correct since for any $m \in \mathcal{M}$:

$$\begin{aligned}
&\Pr\left[\mathsf{Dec}((k_0,k_1), \mathsf{Enc}((k_0,k_1), m)) = m\right] && (12)\\
&= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^{|m|}}\left[\mathsf{Dec}((k_0,k_1), (\mathsf{Enc}_0(k_0,m_0), \mathsf{Enc}_1(k_1, m_0 \oplus m))) = m\right] && (13)\\
&= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^{|m|}}\left[\mathsf{Dec}_0(k_0, \mathsf{Enc}_0(k_0,m_0)) \oplus \mathsf{Dec}_1(k_1, \mathsf{Enc}_1(k_1, m_0 \oplus m)) = m\right] && (14)\\
&= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^{|m|}}\left[m_0 \oplus m_0 \oplus m = m\right] && (15)\\
&= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^{|m|}}\left[m = m\right] && (16)\\
&= 1 && (17)
\end{aligned}$$

□

(b) (1.25 pts) Assuming that Enc_1 is IND-CPA secure, show that Enc is IND-CPA secure (justify all equations and details all steps).
NB: to save typing, you can name your intermediate libraries, number lines like (20) (just use numbers greater than 18 to avoid naming clash) and reuse this number instead of rewriting the whole line.

Correction. To prove that Enc is IND-CPA secure, we need to show that $\mathcal{L}_{\text{cpa-L}} \approx \mathcal{L}_{\text{cpa-R}}$:

$\mathcal{L}_{\text{cpa-L}} \equiv$
$$\boxed{\begin{array}{l}k \leftarrow \mathsf{Gen}(1^\lambda) \\ \underline{\text{EAVESDROP}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \texttt{err} \\ \quad c := \mathsf{Enc}(k, m_L) \\ \quad \text{return } c\end{array}}$$
(Definition $\mathcal{L}_{\text{cpa-L}}$)

$\equiv$
$$\boxed{\begin{array}{l}k_0 \leftarrow \mathsf{Gen}_0(1^\lambda) \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \underline{\text{EAVESDROP}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \texttt{err} \\ \quad m_0 \xleftarrow{\$} \{0,1\}^{|m_L|} \\ \quad m_1 := m_0 \oplus m_L \\ \quad c_0 := \mathsf{Enc}_0(k, m_0) \\ \quad c_1 := \mathsf{Enc}_1(k, m_1) \\ \quad \text{return } (c_0, c_1) \text{ a}\end{array}}$$
(Def Gen and Enc)

$\equiv$
$$\boxed{\begin{array}{l}\mathcal{L}_0 \\ \hline k_0 \leftarrow \mathsf{Gen}_0(1^\lambda) \\ \underline{\text{EAVESDROP}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \texttt{err} \\ \quad m_0 \xleftarrow{\$} \{0,1\}^{|m_L|} \\ \quad c_0 := \mathsf{Enc}_0(k, m_0) \\ \quad c_1 := \text{EAVESDROP}(m_0 \oplus m_L, m_0 \oplus m_R) \\ \quad \text{return } (c_0, c_1)\end{array}} \diamond \mathcal{L}_{\text{cpa-L}}^{\mathsf{Gen}_1, \mathsf{Enc}_1}$$
(Externalize)

$\approx \mathcal{L}_0 \diamond \mathcal{L}_{\text{cpa-R}}^{\mathsf{Gen}_1, \mathsf{Enc}_1}$ (Enc_1 is IND-CPA secure)

$$\equiv \boxed{\begin{array}{l} k_0 \leftarrow \mathsf{Gen}_0(1^\lambda) \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \underline{\text{EAVESDROP}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \mathtt{err} \\ \quad m_0 \xleftarrow{\$} \{0,1\}^{|m_R|} \\ \quad m_1 := m_0 \oplus m_R \\ \quad c_0 := \mathsf{Enc}_0(k, m_0) \\ \quad c_1 := \mathsf{Enc}_1(k, m_1) \\ \quad \text{return } (c_0, c_1) \end{array}} \qquad \text{(Inline and } |m_L| = |m_R| \text{ after first condition)}$$

$$\equiv \mathcal{L}_{\text{cpa-R}} \qquad \text{(Def Enc and Gen)}$$

□

(c) (0.75 pts) For any $m \in \{0,1\}^n$, prove that the following distribution is a uniform distribution over $S := \{(m'_0, m'_1) \in \{0,1\}^n \times \{0,1\}^n \mid m'_0 \oplus m'_1 = m\}$, i.e. for any $(m'_0, m'_1) \in \{0,1\}^n \times \{0,1\}^n$ such that $m = m'_0 \oplus m'_1$:

$$\Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,(m_0, m_1) = (m'_0, m'_1)\,] = \frac{1}{2^n} \tag{18}$$

Similarly, prove that

$$\Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,(m_1, m_0) = (m'_0, m'_1)\,] = \frac{1}{2^n} \tag{19}$$

and conclude that

share-1			share-2		
(14)	$\underline{\text{SAMPLE}(m):}$	$\equiv$	(14)	$\underline{\text{SAMPLE}(m):}$	
(15)	$m_0 \xleftarrow{\$} \{0,1\}^n$		(15)	$m_0 \xleftarrow{\$} \{0,1\}^n$	
(16)	$m_1 := m_0 \oplus m$		(16)	$m_1 := m_0 \oplus m$	
(17.1)	**return** (m_0, m_1)		(17.2)	**return** (m_1, m_0)	

(20)

Correction. Let $m \in \{0,1\}^n$, and $(m'_0, m'_1) \in \{0,1\}^n \times \{0,1\}^n$ such that

$$m = m'_0 \oplus m'_1 \tag{21}$$

Then:

$$\begin{aligned} \Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,(m_0, m_1) = (m'_0, m'_1)\,] &= \Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,m_0 = m'_0 \wedge m_0 \oplus m = m'_1\,] \\ &\qquad \text{(Definition equality on tupple)} \\ &= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^n} [\,m_0 = m'_0 \wedge m_0 \oplus m = m'_1\,] \quad \text{(Definition } m_1\text{)} \\ &= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^n} [\,m_0 = m'_0\,] \\ &\quad (m'_0 \oplus m = m'_1 \text{ true since } m'_0 \oplus m'_1 = m \text{ by assumption}) \\ &= \frac{1}{2^n} \qquad (m_0 \text{ is uniformly sampled}) \end{aligned}$$

For the similar equation, two methods. Method 1:

$$\begin{aligned}
\Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,(m_1, m_0) = (m'_0, m'_1)\,] &= \Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,m_1 = m'_0 \wedge m_0 = m'_1\,] && \text{(Equality on tupple)} \\
&= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^n} [\,m_0 \oplus m = m'_0 \wedge m_0 = m'_1\,] && \text{(Def. } m_1\text{)} \\
&= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^n} [\,m'_1 \oplus m = m'_0 \wedge m_0 = m'_1\,] \\
&\text{(The equation is true iff } m_0 = m'_1\text{, so we can replace } m_0 \text{ with } m'_1\text{)} \\
&= \Pr_{m_0 \xleftarrow{\$} \{0,1\}^n} [\,m_0 = m'_1\,] \\
&(m'_1 \oplus m = m'_0 \text{ is always true since } m = m'_0 \oplus m'_1 \text{ by assumption}) \\
& && (22)
\end{aligned}$$

Or method 2:

$$\Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,(m_1, m_0) = (m'_0, m'_1)\,] = \Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,(m_0, m_1) = (m'_1, m'_0)\,]$$

$$((a,b) = (c,d) \text{ iff } (b,a) = (d,c))$$

but $m' := (m'_1, m'_0) \in S$ since $m'_1 \oplus m'_0 = m$ (eq. (21)), and we just proved before that for any $m' \in S$, $\Pr_{\substack{m_0 \xleftarrow{\$} \{0,1\}^n \\ m_1 := m_0 \oplus m}} [\,(m_1, m_0) = (m'_0, m'_1)\,] = \frac{1}{2^n}$, concluding the proof.
The equivalence between the two libraries is now trivial, since share-1 corresponds to the first studied distribution, and share-2 corresponds to the second distribution, shown to be equal to the first one, hence indistinguishable. □

(d) (1 pts) Assuming that Enc_0 is IND-CPA secure, show that Enc is IND-CPA secure.
NB: to save typing, you can apply the same advice as in the above proof involving Enc_1, and you can also skip the externalize-replace-inline operations by only writing the starting and ending libraries, and **quickly describing all skipped steps**.

Correction. The proof is similar to the security proof assuming that Enc_0 is IND-CPA secure, except that we first use eq. (20) to exchange m_0 and m_1, allowing us to apply then the same

reasoning as before. More formally:

$$\mathcal{L}_{\text{cpa-L}} \equiv \boxed{\begin{array}{l} k_0 \leftarrow \mathsf{Gen}_0(1^\lambda) \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \underline{\textsc{eavesdrop}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \texttt{err} \\ \quad m_0 \xleftarrow{\$} \{0,1\}^{|m_L|} \\ \quad m_1 := m_0 \oplus m_L \\ \quad c_0 := \mathsf{Enc}_0(k, m_0) \\ \quad c_1 := \mathsf{Enc}_1(k, m_1) \\ \quad \text{return } (c_0, c_1) \end{array}} \qquad \text{(Def Gen and Enc)}$$

$$\equiv \boxed{\begin{array}{c} \mathcal{L}_1 \\ \hline k_0 \leftarrow \mathsf{Gen}_0(1^\lambda) \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \underline{\textsc{eavesdrop}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \texttt{err} \\ \quad (m_0, m_1) \leftarrow \textsc{sample}(m_L) \\ \quad c_0 := \mathsf{Enc}_0(k, m_0) \\ \quad c_1 := \mathsf{Enc}_1(k, m_1) \\ \quad \text{return } (c_0, c_1) \end{array}} \diamond \text{sample-1} \qquad \text{(Externalize)}$$

$$\equiv \mathcal{L}_1 \diamond \text{sample-2} \qquad \text{(Previous question)}$$

$$\equiv \boxed{\begin{array}{l} k_0 \leftarrow \mathsf{Gen}_0(1^\lambda) \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \underline{\textsc{eavesdrop}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \texttt{err} \\ \quad m_1 \xleftarrow{\$} \{0,1\}^{|m_L|} \\ \quad m_0 := m_1 \oplus m_L \\ \quad c_0 := \mathsf{Enc}_0(k, m_0) \\ \quad c_1 := \mathsf{Enc}_1(k, m_1) \\ \quad \text{return } (c_0, c_1) \end{array}} \qquad \text{(Inline)}$$

We remark now that this is exactly like the case where Enc_1 was assumed to be IND-CPA, except that now m_0 plays the role of m_1 and Enc_0 plays the role of Enc_1: we can therefore as before externalize the encryption of Enc_0 with $\textsc{eavesdrop}(m_1 \oplus m_L, m_1 \oplus m_R)$ and $\mathcal{L}^{\mathsf{Gen}_1,\mathsf{Enc}_1}_{\text{ind-cpa-L}}$, exchange $\mathcal{L}^{\mathsf{Gen}_1,\mathsf{Enc}_1}_{\text{ind-cpa-R}}$ with $\mathcal{L}^{\mathsf{Gen}_1,\mathsf{Enc}_1}_{\text{ind-cpa-R}}$ (possible since Enc_1 is IND-CPA secure), and we inline again the source code. This gives us:

$$\mathcal{L}_{\text{cpa-L}} \approx \boxed{\begin{array}{l} k_0 \leftarrow \mathsf{Gen}_0(1^\lambda) \\ k_1 \leftarrow \mathsf{Gen}_1(1^\lambda) \\ \underline{\textsc{eavesdrop}(m_L, m_R \in \mathcal{M}):} \\ \quad \text{if } |m_L| \neq |m_R| \text{ return } \texttt{err} \\ \quad m_1 \xleftarrow{\$} \{0,1\}^{|m_R|} \\ \quad m_0 := m_1 \oplus m_R \\ \quad c_0 := \mathsf{Enc}_0(k, m_0) \\ \quad c_1 := \mathsf{Enc}_1(k, m_1) \\ \quad \text{return } (c_0, c_1) \end{array}} \qquad \text{(Inline)} \tag{23}$$

We have now exactly the same code as in eq. (Inline), except that m_L is replaced with m_R. We can therefore apply exactly the same operation as before but in reverse (externalize sample-2, replace with sample-1) to recover:

$$\mathcal{L}_{\text{cpa-L}} \approx \mathcal{L}_{\text{cpa-R}} \tag{24}$$

concluding the proof. □