

# Cryptography 101: Chosen ciphertext attacks

ComComX • 3 May 2026

In the spirit of the Kerckhoff's principle, we consider the question: is security of an encryption scheme ( $\text{KeyGen}$ ,  $\text{Enc}$ ,  $\text{Dec}$ ) is still possible if the adversary has access to the decryption function  $\text{Dec}$ , i.e., she can read the decryptions of any ciphertexts of her choice. This is well known as *chosen ciphertext attack*, and the security under such kind of attack can be captured by the following security game:

IND-CCA2( $\lambda$ ): (Indistinguishability under CCA)

1. Cha samples  $b \leftarrow U_1$  and  $k \leftarrow \text{KeyGen}(1^\lambda)$
2. For  $i = 1, \dots, \ell$ , for some  $\ell = \text{poly}(\lambda)$ :
  - Adv chooses some message  $\hat{m}_i$  and sends to Cha
  - Cha sends back  $\text{Enc}(1^\lambda, k, \hat{m}_i)$  to Adv
  - Adv chooses some ciphertext  $\hat{c}_i$  and sends to Cha
  - Cha sends back  $\text{Dec}(1^\lambda, k, \hat{c}_i)$  to Adv
3. Adv chooses two message  $m_0, m_1$  and sends to Cha
4. Cha sends back  $c = \text{Enc}(1^\lambda, k, m_b)$  to Adv
5. For  $i = 1, \dots, \text{poly}(\lambda)$ :
  - Adv chooses some message  $\hat{m}_{\ell+i}$  and sends to Cha
  - Cha sends back  $\text{Enc}(1^\lambda, k, \hat{m}_{\ell+i})$  to Adv
  - Adv chooses some ciphertext  $\hat{c}_{\ell+i} \neq c$  and sends to Cha
  - Cha sends back  $\text{Dec}(1^\lambda, k, \hat{c}_{\ell+i})$  to Adv
6. Adv guesses a bit  $b'$
7. Adv wins if  $b' = b$ .

**Definition 1** (IND-CCA security). *An encryption scheme ( $\text{KeyGen}$ ,  $\text{Enc}$ ,  $\text{Dec}$ ) is CCA-secure if, for any PPT algorithm  $A$ ,*

$$\Pr[A \text{ wins IND-CCA2}(\lambda)] = \frac{1}{2} + \text{negl}(\lambda).$$

Note: without step 5, the game is called IND-CCA1( $\lambda$ ) game and the corresponding security notion is CCA1 security.

Recall the CPA-secure encryption scheme based on PRF:

- $\text{KeyGen}_{CPA}(1^\lambda)$ :
  1. Sample  $k \leftarrow U_\lambda$
  2. Output  $f_k$
- $\text{Enc}_{CPA}(1^\lambda, k, m)$ :
  1. Pick  $i \leftarrow \{0, 1\}^\lambda$
  2.  $c = m \oplus f_k(i)$
  3. Output  $(i, c)$
- $\text{Dec}_{CPA}(1^\lambda, k, (i, m))$ :
  1.  $\hat{m} = c \oplus f_k(i)$
  2. Output  $\hat{m}$ .

**Claim 2.** *The above CPA-secure encryption scheme is not CCA-secure.*

*Proof.* With the PRF-based encryption scheme, we can construct an adversary to exploit the decryption function to compute  $m_b$  exactly as follows:

- In step 4 of the game IND-CCA2, our adversary receives a ciphertext  $c = \text{Enc}(1^\lambda, k, m_b) = (x, m_b \oplus f_k(x))$  from Cha
- Sample  $r \leftarrow U_\lambda$  and  $r \neq 0$
- Compute  $z = m_b \oplus f_k(x) \oplus r$
- Query the decryption of  $\hat{c} = (y, z)$ . Clearly,  $\hat{c} \neq c$
- Adversary receives the decryption
  - $\hat{m} = f_k(y) \oplus z = f_k(y) \oplus m_b \oplus f_k(y) \oplus r = m_b \oplus r$
  - Compute  $m = \hat{m} \oplus r$ .

Then, the adversary can compute  $b$  by comparing  $m$  to  $m_0$  and  $m_1$ , and breaks the security with probability 1.  $\square$

**Theorem 3.** Let  $\Pi = (\text{KeyGen}_{CPA}, \text{Enc}_{CPA}, \text{Dec}_{CPA})$  be a CPA-secure encryption scheme. Let  $\Sigma = (\text{Gen}, \text{Mac}, \text{Verify})$  be a EUF-CMA-secure MAC scheme. Then the following encryption-then-mac scheme is CCA-secure:

KeyGen( $1^\lambda$ ):

1. Sample  $k_1 \leftarrow \text{KeyGen}_{CPA}(1^\lambda)$  and  $k_2 \leftarrow \text{Gen}(1^\lambda)$
2. Output  $(k_1, k_2)$

Enc( $1^\lambda, (k_1, k_2), m$ ):

1. Run  $c \leftarrow \text{Enc}_{CPA}(1^\lambda, k_1, m)$
2. Run  $t \leftarrow \text{MAC}(1^\lambda, k_2, c)$
3. Output  $(c, t)$

Dec( $1^\lambda, (k_1, k_2), (c, t)$ ):

1. If  $\text{Verify}(1^\lambda, k_2, c, t) = 1$  then output  $\text{Dec}_{CPA}(1^\lambda, k_1, c)$
2. Otherwise, output  $\perp$ .

*Proof.* Let first recall the IND-CPA game used to define the CPA-security:

IND-CPA( $\lambda$ ):

1. Cha chooses a random bit  $b \leftarrow \{0, 1\}$  and key  $k \leftarrow \text{KeyGen}(1^\lambda)$
2. For  $i = 1, \dots, t$  (with  $t = \text{poly}(\lambda)$ ):
  - Adv chooses  $\hat{m}_i \in \mathcal{M}$  and sends it to Cha
  - Cha sends back  $\text{Enc}(1^\lambda, k, \hat{m}_i)$
3. Adv chooses  $m_0, m_1$  and sends to Cha
4. Cha sends back  $\text{Enc}(1^\lambda, k, m_b)$
5. Adv guesses a bit  $b'$
6. Adv wins if  $b' = b$ .

With  $\Pi$  and  $\Sigma$ , the game IND-CCA2 is played as follows:

IND-CCA2( $\lambda$ ):

1. Cha samples  $b \leftarrow U_1$  and  $(k_1, k_2) \leftarrow \text{KeyGen}(1^\lambda)$
2. For  $i = 1, \dots, \ell$ , for some  $\ell = \text{poly}(\lambda)$ :
  - Adv chooses some message  $\hat{m}_i$  and sends to Cha
  - Cha computes  $c_i = \text{Enc}_{CPA}(1^\lambda, k_1, \hat{m}_i)$  and sends back
$$(c_i, t_i) = (c_i, \text{MAC}(1^\lambda, k_2, c_i))$$
  - Adv chooses some ciphertext  $(\hat{c}_i, \hat{t}_i)$  and sends to Cha
  - If  $\text{Verify}(1^\lambda, k_2, \hat{c}_i, \hat{t}_i) = 1$  then Cha returns  $\text{Dec}_{CPA}(1^\lambda, k_1, \hat{c}_i)$
  - Otherwise, Cha returns  $\perp$
3. Adv chooses two message  $m_0, m_1$  and sends to Cha
4. Cha sends back  $(c, t) = (\text{Enc}_{CPA}(1^\lambda, k_1, m_b), \text{MAC}(1^\lambda, k_2, c))$
5. For  $i = 1, \dots, \text{poly}(\lambda)$ :
  - Adv chooses some message  $\hat{m}_{\ell+i}$  and sends to Cha
  - Cha computes  $c_i = \text{Enc}_{CPA}(1^\lambda, k_1, \hat{m}_{\ell+i})$  and sends back
$$(c_{\ell+i}, t_{\ell+i}) = (c_{\ell+i}, \text{MAC}(1^\lambda, k_2, c_{\ell+i}))$$
  - Adv chooses some ciphertext  $(\hat{c}_{\ell+i}, \hat{t}_{\ell+i}) \neq (c, t)$  and sends to Cha
  - If  $\text{Verify}(1^\lambda, k_2, \hat{c}_{\ell+i}, \hat{t}_{\ell+i}) = 1$  then Cha returns  $\text{Dec}_{CPA}(1^\lambda, k_1, \hat{c}_{\ell+i})$
  - Otherwise, Cha returns  $\perp$
6. Adv guesses a bit  $b'$
7. Adv wins if  $b' = b$ .

Now, assume for contradiction that the encryption-then-mac scheme is not CCA-secure. That is, there is a PPT adversary  $A$  that wins the above game IND-CCA2 with noticeable advantage over random guessing. We use  $A$  to construct an adversary to either break  $\Pi$  or  $\Sigma$ .

We consider an event  $\mathcal{E}$ : at some point in the IND-CCA2 game,  $A$  submits a decryption query  $(c, t)$  that is a successful existential forgery, i.e.,  $\text{Verify}(k_2, c, t) = 1$  and  $(c, t)$  was never produced by the encryption oracle. We consider two possibilities:

- $\mathcal{E}$  occurs with noticeable probability: in this case, we construct an adversary  $B$  to break the EUF-CMA security of  $\Sigma$ .

- $\mathcal{E}$  occurs with negligible probability: in this case, we construct an adversary  $D$  to break the CPA security of  $\Pi$ .

**Case 1:**  $\mathcal{E}$  occurs with noticeable probability. This means that  $A$  is capable of forging a valid tag for a new ciphertext. Let  $B$  play the EUF-CMA game while taking the role of  $A$ 's Cha in IND-CCA2 as follows:

- $B$  samples  $b$  and  $k_1 \leftarrow \text{keyGen}_{CPA}(1^\lambda)$
- Every time the MAC function is called by  $A$ :  $B$  forwards the query to its Cha, gets the returned tag, and forwards to  $A$
- When  $A$  submits a decryption query  $(c, t)$ ,  $B$  checks if it is a "fresh" query (i.e.,  $t$  was not produced by  $B$ 's Cha before)
- As soon as  $A$  submits a fresh query  $(c, t)$  that is valid,  $B$  outputs  $(c, t)$  as its own forgery to its Cha.

Since  $A$  is playing exactly an instance of IND-CCA2, there is a noticeable that event  $\mathcal{E}$  happens. It means that  $B$  win EUF-CMA with noticeable probability, breaking the EUF-CMA-security of  $\Sigma$ .

**Case 2:**  $\mathcal{E}$  occurs with negligible probability. This means that  $A$  almost never produces a valid decryption query that was not produced by the encryption oracle. In this case, we let  $D$  play the game IND-CPA while taking the role of  $A$ 's Cha in IND-CCA2 as follows:

- $D$  samples  $k_2 \leftarrow \text{Gen}(1^\lambda)$
- Every time  $A$  make an encryption query for message  $m_i$ :
  - $D$  requests an encryption  $c_i = \text{Enc}_{CPA}(1^\lambda, k_1, m_i)$  from its cha
  - $D$  computes  $t_i = \text{MAC}(1^\lambda, k_2, c_i)$
  - $D$  returns  $(c_i, t_i)$  to  $A$
- Every time  $A$  makes a decryption query,  $D$  just returns  $\perp$
- When  $A$  sends  $m_0, m_1$ :
  - $D$  sends these message to its Cha and receives  $c = \text{Enc}_{CPA}(1^\lambda, k_1, m_b)$
  - $D$  computes  $t = \text{MAC}(1^\lambda, k_2, c_b)$
  - $D$  returns  $(c, t)$  to  $A$
- $D$  outputs whatever outputted by  $A$ .

We have that

$$\begin{aligned}\Pr[D \text{ wins IND-CPA}(\lambda)] &= \Pr[D \text{ wins IND-CPA}(\lambda) \wedge \bar{\mathcal{E}}] \\ &= \Pr[A \text{ wins IND-CPA}(\lambda) \wedge \bar{\mathcal{E}}] \\ &= \Pr[A \text{ wins IND-CPA}(\lambda)] - \Pr[A \text{ wins IND-CPA}(\lambda) \wedge \mathcal{E}] \\ &\geq \Pr[A \text{ wins IND-CPA}(\lambda)] - \Pr[\mathcal{E}] \\ &\geq \frac{1}{2} + \frac{1}{p(\lambda)} - \text{negl}(\lambda)\end{aligned}$$

for some polynomial  $p$ . This probability is noticeably better than  $1/2$ , thus  $D$  successfully breaks the CPA-security of  $\Pi$ .

In both cases, the our assumption leads to contradictions. Therefore, the encryption-then-mac scheme must be CCA-secure.  $\square$