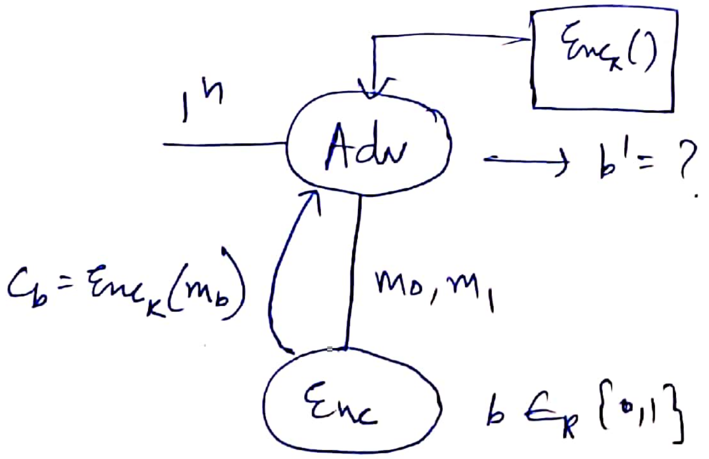
CPA-Secure Encryption Scheme

No deterministic encryption scheme is CPA-secure.

In CPA attack, the attacker can select two different plaintexts, say m_0, m_1 and fetch their encryptions, say c_0 and c_1 respectively. If the encryption scheme is deterministic, the attacker can compare the ciphertext got from the channel with that fetched from the encryption server it has access to and hence determine which plaintext the ciphertext corresponds to.

Formally, $\forall$ PPTM A, ~~A chooses~~ for two plaintexts m_0, m_1 and their encryptions c_0, c_1, A receives encryption c_b of m_b where $b \in_R \{0,1\}$.

The adversary sends m_0 and m_1 for encryption and compares it with c_b. If $c_b == c_0$, $b = 0$ & $m_b = m_0$ ~~and~~, otherwise $b = 1$ and $m_b = m_1$.



Therefore, for an encryption scheme to achieve CPA-security, it must use some randomness in the encryption process to ensure that the same plaintext doesn't always get mapped to the same ciphertext.

Let r be selected at random. The encryption scheme goes as follows: —

$$Enc_k(m) = c = F_k(r) \oplus m$$

where m is the message, k is the key and F is a pseudo-random function.

The tuple $<r, c>$ is sent through the public channel. The decryption goes as follows:

$$\begin{aligned} Dec_k(c) &= c \oplus F_k(r) \\ &= F_k(r) \oplus m \oplus F_k(r) \\ &= m \end{aligned}$$

From the definition of pseudo-random function, it is possible to distinguish between the encrypted message and a random message with less than negl(n) probability. Due to this randomness (pseudo-randomness), the PPTM attacker cannot make out which plaintext the received ciphertext corresponds to. This makes the scheme CPA-secure.

Mode of Operation : CTR

The messages are divided into message blocks of equal size and encrypted in certain modes of operation. We describe the Randomized Counter mode here.
(padding : $1(0)^*$)

Given message $m = \langle m_1, m_2, \ldots, m_t \rangle$ and random r, the encryption scheme goes as follows:

$$r_i = F_k(r+i)$$

$$Enc_k(m_i) = c_i = F_k(r_i) \oplus m_i$$

for $i \in \{0, 1, \ldots, t\}$ and $c_0 = r$.

The ciphertext $c = c_0 \| c_1 \| \ldots \| c_t$.

Decryption: $r_i = F_k(r+i)$

$$Dec_k(c_i) = F_k(r_i) \oplus c_i = F_k(r_i) \oplus m_i \oplus F_k(r_i) = m_i$$

The original message is calculated as

$m = m_1 \| m_2 \| \ldots \| m_t$ as intended.

CTR mode also takes care of not revealing message information for repeating blocks since a different "r" is used for each block.