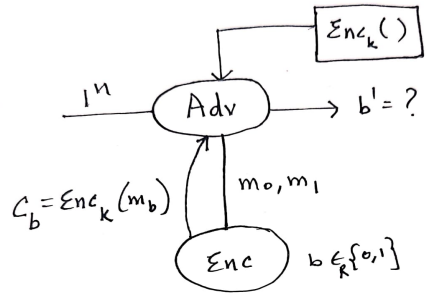
## CPA-Secure Encryption Scheme

*No deterministic encryption scheme is CPA-secure.*
In CPA attack, the attacker can select two different plaintexts, say $m_0$, $m_1$ and ask for 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 got from the encryption server it has accesss to and hence determine which plaintext the ciphertext corresponds to.
Formally, $\forall$ PPTH adversary A, A chooses two plaintexts $m_0$ and $m_1$ and receives the encryption $c_b$ of $m_b$ where $b \in \{0, 1\}$ and is selected randomly. The adversary has access to the encryption server. It send $m_0$ and $m_1$ to the server and gets their corresponsing ciphertexts $c_0$ and $c_1$. If $c_b == c_0$, then $b = 0$, else $b = 1$.




Therefore, for a encryption scheme to achieve CPA-security, it must use some form of randomness in the encryption process to ensure that the same plaintext doesn't always map 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 to be encrypted, k is the secret key and F is a pseudo-random function. The tuple $< r, c >$ is sent through the public channel. The decryption goes as follows:

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

From the definition of pseudo-random function, it is possible to distinguish between the encrypted message and random message with less than negligible probability. Due to this randomness, each time the attacker gets an encryption of the same message, the ciphertexts will appear random to them. Hence, the attacker won't be able to make out which plaintext that particular ciphertext corresponds to. This makes the scheme CPA-secure.
In practice, messages are divided into message blocks of equal size and encrypted in certain modes of operation. We describe the Randomized Counter Mode here. Given message $m =< m_1, m_2, \ldots m_t >$ and random value r, the encryption scheme goes as follows:

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

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

$\forall i \in 0, 1, \ldots t$ and $c_0 = r$. The encrypted message is $c = c_0||c_1||\ldots||c_t$. During decryption, the random seed r is fetched from ciphertext as $c_0 = r$.

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

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

The original message is computed as $m = m_1||m_2||\ldots||m_t$ as intended.