Chosen Ciphertext attack (CCA):

In this attack, we provide the adversary with the ability to encrypt any messages of its choice as in a chosen-plaint text attack (giving access to encryption server), also provide the adversary with the ability to decrypt any ciphertexts of its choice (giving access to decryption server).

The CCA indistinguishability experiment:

- A random key k is generated by running $Gen(1^n)$.
- Adversary A is given input 1^n and access to encryption server and decryption server. It outputs a pair of messages m_0, m_1 of the same length.
- A random bit $b \leftarrow \{0,1\}$ is chosen, and then a ciphertext $c \leftarrow Enc_k(m_b)$ is computed and given to A. We call this a challenge ciphertext.
- The adversary A continues to have oracle access to encryption server, and decryption server, but he is not supposed to ask decryption of challenge ciphertext and A outputs a bit b'
- If b'= b, then we say Adversary succeed in experiment and the output of experiment is 1 otherwise 0.

Why does the CPA Encryption scheme fail here?

$$c = F_k(r)\ XOR\ m$$

Let us say we have two messages m_1, m_2 and c_1, c_2 . The first message contains all 0's and the second message contain all 1's. If you toggle one of the ciphertext MSB and decryption of it will tell you the original message which is used to encrypt. If the cipher text is 10^{n-1} then it is the message which contains all zeros or if the cipher text is 01^{n-1} then it is the message which contains all ones. So, this scheme is not secure under CCA attack.

So, we need a mechanism which should not help to know changes on plaintext when adversary changes ciphertext. We need non-mailable encryption schemes here.

To Achieve CCA level security we use MAC.

CCA-Secure Encryption:

Here we first encrypt the message and then will apply MAC on that ciphertext.

Let $\pi_E = (Gen_E, Enc, Dec)$ be a CPA-Secure Encryption scheme and $\pi_M = (Gen_M, Mac, Vrfy)$ a secure message authentication code. Then we define CCA-secure encryption like below

Define a CCA-secure encryption scheme as follows:

- $\mathsf{Gen}'(1^n)$: upon input 1^n, choose $k_1, k_2 \leftarrow \{0,1\}^n$
- $\mathsf{Enc}'_k(m)$: upon input key (k_1, k_2) and plaintext message m, compute $c = \mathsf{Enc}_{k_1}(m)$ and $t = \mathsf{Mac}_{k_2}(c)$ and output the pair (c, t)
- $\mathsf{Dec}'_k(c, t)$: upon input key (k_1, k_2) and ciphertext (c, t), first verify that $\mathsf{Vrfy}_{k_2}(c, t) = 1$. If yes, then output $\mathsf{Dec}_{k_1}(c)$; if no, then output $\perp$.

Here if Adversary try to change the ciphertext then MAC verification will fail, so decryption server useless here.MAC ensures that there are no changes in ciphertext.