(5) Use aforementioned CPA security & secure MCA to design a provably CCA secure encryption scheme.

why we need CCA when CPA is already secure?

Let's take a scenario

- Alice wants to login on Bob's server.
- Alice browser has a shared key k with bob.
- Alice encrypts password with k & sends

$$c = Enc_k(password)$$

- Bob decrypts c & if password is correct it allows Alice to login.

There might be attack possible that a resetter algorithm $Res(c, i, b) \to c'$, c is encrypted by $m = m_1 \cdots m_k$

i is location of m which will be changed, b is the change bit. $Dec(c') \to m_1 \cdots m_i \cdots m_k$ where $m_i = b$ attacker don't even know o/p after producing c'. But they can know which m be changed from feedback of password verification. So using that, attacker can recover the password for at most 100 tries by using the function. So only CPA is not totally secure.

Achieve CCA security using MAC with CPA

→ we will use encryption then authentica^n as CCA security.

To Proof:- In order to prove private-key encryption scheme is CCA secure, intitution is to prove Dec() provided for adversary is useless, back to CPA security.

Assumptions

- private-key (Enc, Dec) scheme which is CPA security
- MAC ~~alg~~ authentica^n algorithm (MAC, Vrfy) which has stronger security, meaning that for every message MAC tag is unique.

Proof:- If we can show that if adversary A can break CCA security of new encryption scheme with polynomial time then using A, we can break either the above assumption

(a) Attempt for Breaking MAC

if $Pr[E] \geq \varepsilon/2$, then $\varepsilon/2$ will be negligible so, we can get attacker B_1 who can break security of MAC with probability $\varepsilon/2$

B_1 will simulate a copy of CPA in its head. Then run A & tries to provide what A needs. It will use CPA & MAC to implement (supposedly CCA secure encryption scheme. The first moment that A makes that event E happen, it means that it has found ciphertext $c' = [c, t]$ when t is correct tag for c & A has not obtained this c' from B'.

(b) Attempt for breaking CPA encryption

If $Pr[E] \leq \varepsilon$, it means that probability at least ε, A succeeds without asking any query like c' described above. It means that A asks for decryption is something that it knows the answer already. So if we remove decryption from A, it can still win in CPA security with probability of at least $\varepsilon/2$.

By both of these attempt, we can say that CCA is secure formed by CPA & MAC.