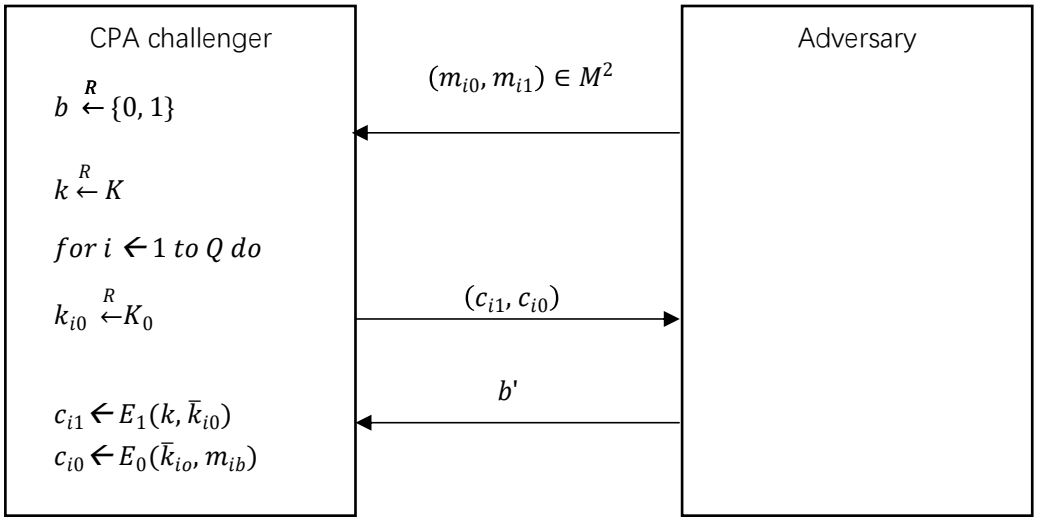
*1.a) Attack Game 0*

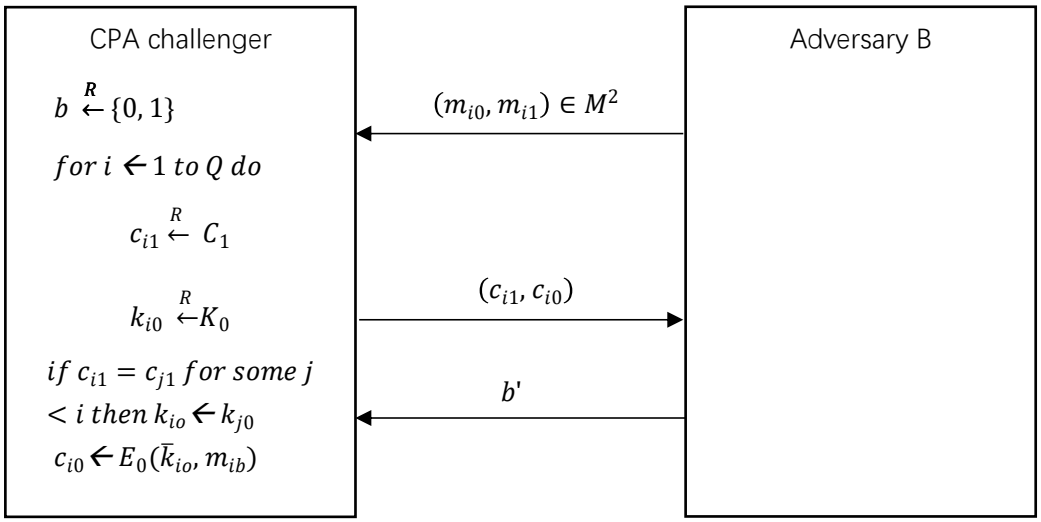
For a given hybrid cipher $\varepsilon = (E, D)$, defined over $(K, M, C_0 \times C_1)$ and for a given adversary A, we define two experiments, Experiments 0 and Experiments 1. For b = 0, 1, we define Experience b.




$$CPAadv^*[A, \varepsilon] = \left|\Pr[W_0] - \frac{1}{2}\right|, W_0\ is\ the\ event\ b = b'$$

*Attack Game 1*

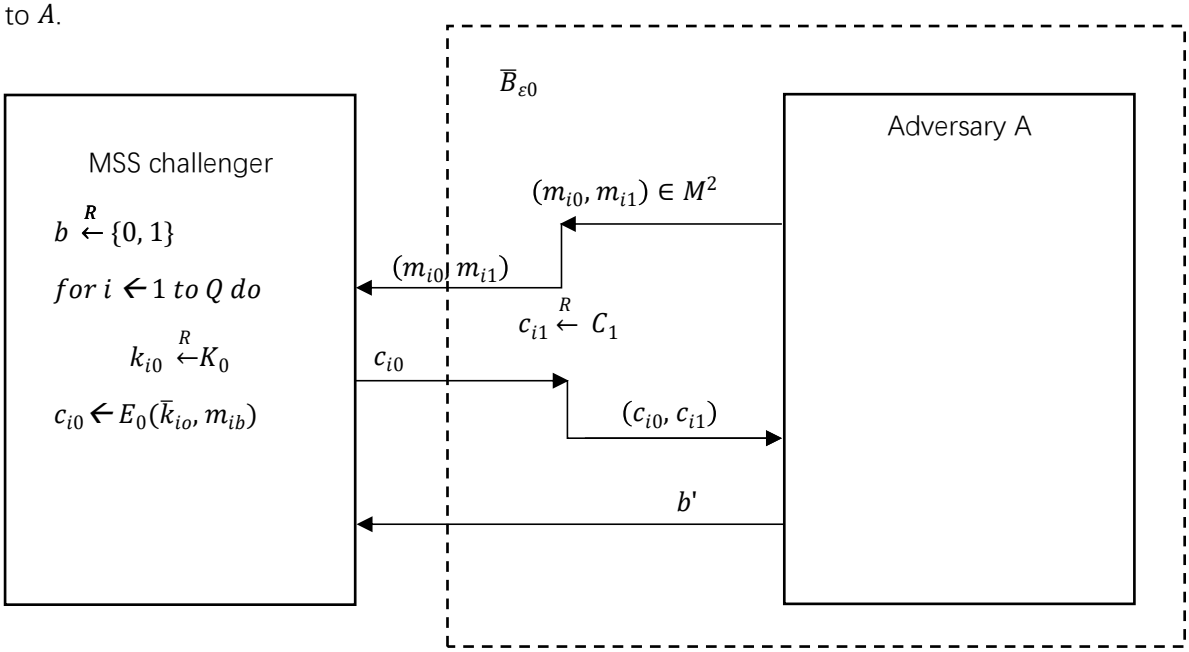
We know cipher $\varepsilon_1 = (E_1, D_1)$ is a CPA secure cipher defined over $(K, K_0, C1)$, so the value $CPAadv(B, \varepsilon_1)$ is $'negligible'$, replace $c_{i1} \leftarrow E1(k, \bar{k}_{i0})$ by $c_{i1} \overset{R}{\leftarrow} C_1$.




$$|\Pr[W_1] - \Pr[W_0]| = CPAadv[B, \varepsilon]$$

***Attack Game 2***

$\overline{B}_{\varepsilon 0}$ is a multi -key semantic security adversary, playing multi -key semantic security attack with $MSS$ challenger. It makes at most $Q$ queries. Adversary $\overline{B}_{\varepsilon 0}$ plays the role of challenger to $A$.




Then

$$\left|\Pr[W_2] - \frac{1}{2}\right| = MSSadv^*[\overline{B}_{\varepsilon 0}, \varepsilon_0]$$

Applying the Difference Lemma, we have

$$|\Pr[W_2] - \Pr[W_1]| \leq \frac{Q^2}{2N}$$

Game 3, independent encryption keys $k_{i0}$ are used to encrypt each message. So,

$$MSSadv^*[\overline{B}_{\varepsilon 0}, \varepsilon_0] = Q \cdot SSadv^*[B_{\varepsilon 0}, \varepsilon_0]$$

Putting together:

$$CPAadv^*[A, \varepsilon] = \left|\Pr[W_0] - \frac{1}{2}\right|$$

$$|\Pr[W_1] - \Pr[W_0]| = CPAadv[B, \varepsilon]$$

$$\left|\Pr[W_2] - \frac{1}{2}\right| = MSSadv^*[\overline{B}_{\varepsilon 0}, \varepsilon_0]$$

$$|\Pr[W_2] - \Pr[W_1]| \leq \frac{Q^2}{2N}$$

$$MSSadv^*[\overline{B}_{\varepsilon 0}, \varepsilon_0] = Q \cdot SSadv^*[B_{\varepsilon 0}, \varepsilon_0]$$

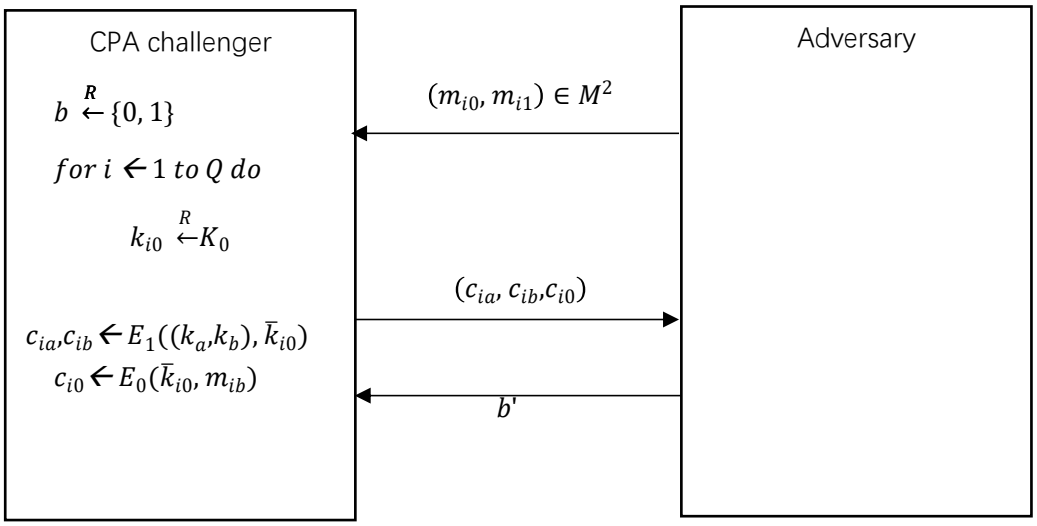
So:

$$CPAadv^*[A, \varepsilon] \leq \frac{Q^2}{2N} + CPAadv[B, \varepsilon] + Q \cdot SSadv^*[B_{\varepsilon 0}, \varepsilon_0]$$

Finally, we prove that this hybrid cipher $\varepsilon = (E, D)$, defined over $(K, M, C_0 \times C_1)$ is CPA secure.

***1.b)***

From the part (a), we know that $\varepsilon_1 = (E_1, D_1)$ is a CPA secure cipher. So, if an eavesdropper can only see $(c_a, c_b, c_0)$, from CPA attack game, we know this eavesdropper can't learn the $k_0$.




We regard $k_a$ and $k_b$ as two keys randomly selected from $K$, We can see that the difference between $\varepsilon'$ and $\varepsilon$ is that $\varepsilon'$ uses $k_a$ and $k_b$. In fact, it is equivalent to doing the encryption scheme of $\varepsilon$ twice. Therefore,

$$CPAadv^*[A, \varepsilon'] \ is\ still\ 'negligible'$$

We know that if the short ciphertext header $(c_a, c_b)$ is encrypted by a CPA secure cipher and the long ciphertext body $c_0$ is encrypted by a semantically secure cipher, the cipher $\varepsilon' = (E', D')$ is CPA secure.

***2.***

***Conditions:***

- Sam must be a trusted third party, recognized by both Alice and Bob.
- Sam can't send r to Alice.
- Sam can't send x or $x_a$ to Bob.
- Sam can't know the $(k_0, k_1)$.

***Problem:***

If $(k_0, k_1)$ are used twice, for $r_1, r_2 \in Z_p$,

$$x_{b1} = r_1(b + k_0) + k_1$$
$$x_{b2} = r_2(b + k_0) + k_1.$$

$x_a = a + k_0$ is same. So,

$$x_1 = r_1 x_a - x_{b1}$$
$$x_2 = r_2 x_a - x_{b2}$$

then

$$x_1 - x_2 = (r_1 - r_2)x_a + (x_{b2} - x_{b1}) = (r_1 - r_2)(a + k_0) + (r_2 - r_1)(b + k_0)$$
$$= (r_1 - r_2)(a - b)$$

We can also get

$$x_2 - x_3 = (r_2 - r_3)(a - b)$$

...

Therefore, if $(k_0, k_1)$ are used more than once, Alice can learn something about b.

***Solution:***

Let $\varepsilon = (E, D)$ is a CPA secure cipher, defined over $(Z_p, K, C)$.

We add a condition:

- *Before Alice and Bob send information each time, Sam chooses a random $k \in K$ and sends $k$ to Alice and Bob.*

By default, Sam does not give Alice and Bob different $k$ in the same protocol process. The new protocol works as follows:

1) Sam chooses random $k \in K$ and sent $k$ to Alice and Bob.
2) Bob and Alice use the CPA secure cipher to encrypt $k$, $c_0 = E(k_0,k), c_1 = E(k_1,k)$.
3) Bob chooses random number $r \in Z_p$, and send $r$, $x_b^{'} = r(b + c_0) + c_1$ to Sam.
4) When Alice wants to test equality, she sends $x_a^{'} = a + c_0$ to Sam.
5) Sam compute $x^{'} = rx_a^{'} - x_b^{'}$ and sends back to Alice.
6) Alice check if $x^{'} + c_1 = 0$