Assignment 3

Submission Deadline: 2021.10.13, 12:00 pm

1. (Hybrid CPA construction). Let (E_0, D_0) be a semantically secure cipher defined over (K_0, M, C_0), and let (E_1, D_1) be a CPA secure cipher defined over (K, K_0, C_1).

   a) Define the following hybrid cipher (E, D) as:

$$E(k,m) := \{k_0 \xleftarrow{\text{R}} \mathcal{K}_0,\ c_1 \xleftarrow{\text{R}} E_1(k,k_0),\ c_0 \xleftarrow{\text{R}} E_0(k_0,m),\ \textbf{output } (c_1,c_0)\}$$
$$D\big(k,\ (c_1,c_0)\big) := \{k_0 \leftarrow D_1(k,c_1),\ m \leftarrow D_0(k_0,c_0),\ \textbf{output } m\}$$

   Here c1 is called the ciphertext header, and c0 is called the ciphertext body. Prove that (E, D) is CPA secure.

   b) Suppose m is some large copyrighted content. A nice feature of (E,D) is that the content owner can make the long ciphertext body c0 public for anyone to download at their leisure. Suppose both Alice and Bob take the time to download c0. When later Alice, who has key ka, pays for access to the content, the content owner can quickly grant her access by sending her the short ciphertext header $c_a \leftarrow E_1(k_a, k_0)$. Similarly, when Bob, who has key kb, pays for access, the content owner grants him access by sending him the short header $c_b \leftarrow E_1(k_b, k_0)$. Now, an eavesdropper gets to see $E'\big((k_a, k_b), m\big) = (c_a, c_b, c_0)$. Generalize your proof from part (a) to show that this cipher is also CPA secure.

2. Assume Alice and Bob wants to run a protocol to compare two numbers from each of them, namely number a from Alice and number b from Bob, here a and b are greater than 0 and less than some big prime number p. Alice wants to know if a=b; but if $a \neq b$ then Alice should learn nothing else about b; Bob should learn nothing at all about a.

   We introduce a trust third party Sam to help by allowing Alice and Bob to interact with the server Sam. Suppose Alice and Bob have a shared secret key $(k_0, k_1) \in Z_p^2$, and Alice and Bob each have a secure channel to Sam. The protocol works as follows:

1) Bob chooses random number $r \in Z_p$, and send $r, x_b = r(b + k_0) + k_1$ to Sam.
2) When Alice wants to test equality, she sends $x_a = a + k_0$ to Sam.
3) Sam computes $x = r x_a - x_b$ and sends back to Alice.
4) Alice check if $x + k_1 = 0$

In order for this protocol to work properly, what conditions do we need to put on Sam? If $k_0,\ k_1$ are used more than once, there could be problems, please explain what trouble will it make and how to prevent it by giving your solution.