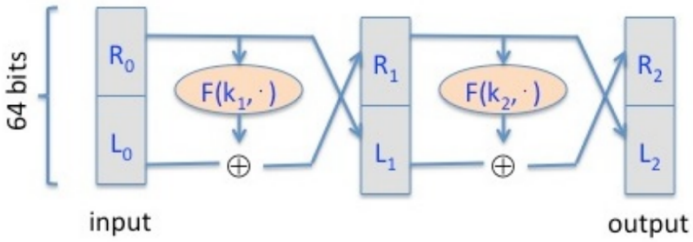
Assignment 2 - 2019.09.24

1. Recall that the Luby-Rackoff theorem discussed states that applying a three round Feistel network to a secure PRF gives a secure block cipher. Let's see what goes wrong if we only use a two round Feistel. Let $F: K \times \{0,1\}^{32} \rightarrow \{0,1\}^{32}$ be a secure PRF. Recall that a 2-round Feistel defines the following PRP $F_2: K^2 \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$



Here R_0 is the right 32 bits of the 64-bit input and L_0 is the left 32 bits. One of the following lines is the output of this PRP F_2 using a random key, while the other three are the output of a truly random permutation $f: \{0,1\}^{64} \rightarrow \{0,1\}^{64}$. All 64-bit outputs are encoded as 16 hex characters. Can you say which is the output of the PRP? Note that since you are able to distinguish the output of F_2 from random, F_2 is not a secure block cipher, which is what we wanted to show.
Hint: First argue that there is a detectable pattern in the xor of $F_2(\cdot, 0^{64})$ and $F_2(\cdot, 1^{32}0^{32})$

1) On input 0^{64} the output is "9d1a4f78 cb28d863". On input $1^{32}0^{32}$ the output is "75e5e3ea 773ec3e6".
2) On input 0^{64} the output is "e86d2de2 e1387ae9". On input $1^{32}0^{32}$ the output is "1792d21d b645c008".
3) On input 0^{64} the output is "2d1cfa42 c0b1d266". On input $1^{32}0^{32}$ the output is "eea6e3dd b2146dd0"
4) On input $0^{64}$the output is "4af53267 1351e2e1". On input $1^{32}0^{32}$the output is "87a40cfa 8dd39154"

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 choose 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 $x = rx_a - x_b$ 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.