Solution of Exercisesheet 10

Exercise 10-1

L_0	R_0
$L_1 = R_0$	$R_1 = L_0 \oplus f_1(R_0)$
$L_2 = R_1 = L_0 \oplus f_1(R_0)$	$R_2 = L_1 \oplus f_2(R_1) = R_0 \oplus f_2(L_0 \oplus f_1(R_0))$
$L_3 = R_2 = R_0 \oplus f_2(L_0 \oplus f_1(R_0))$	$R_3 = L_2 \oplus f_3(R_2) = L_0 \oplus f_1(R_0) \oplus f_3(R_0 \oplus f_2(L_0 \oplus f_1(R_0)))$

$$L_4 = R_3 = L_0 \oplus f_1(R_0) \oplus f_3(R_0 \oplus f_2(L_0 \oplus f_1(R_0)))$$
$$R_4 = R_0 \oplus f_2(L_0 \oplus f_1(R_0)) \oplus f_4(L_0 \oplus f_1(R_0) \oplus f_3(R_0 \oplus f_2(L_0 \oplus f_1(R_0))))$$

Inversion:

L_3	R_3
$L_2 = R_3 \oplus f_3(R_2) = R_3 \oplus f_3(L_3)$	$R_2 = L_3$
$L_1 = R_2 \oplus f_2(R_1) = L_3 \oplus f_2(R_3 \oplus f_3(L_3))$	$R_1 = L_2 = R_3 \oplus f_3(L_3)$
$L_0 = R_1 \oplus f_1(R_0) = R_3 \oplus f_3(L_3) \oplus f_1(L_3 \oplus f_2(R_3 \oplus f_3(L_3)))$	$R_0 = L_1 = L_3 \oplus f_2(R_3 \oplus f_3(L_3))$

1. **To show:** Prove that a two-round Feistel network using pseudorandom round functions is not a pseudorandom permutation.
   The adversary first queries a random string $L_0||R_0$ to the oracle. After two rounds he gets $L_2||R_2 = L_0 \oplus f_1(R_0)||R_0 \oplus f_2(L_0 \oplus f_1(R_0))$ back, if the oracle answers with the PRP. Then he queries a second string $L_0^*||R_0$, where L_0^* is L_0 with the first bit flipped. He then gets $L_2^*||R_2^* = L_0^* \oplus f_1(R_0)||R_0 \oplus f_2(L_0^* \oplus f_1(R_0))$ back, if the oracle answers with the PRP. It is easy to check whether L_2 and L_2^* differ only in the first bit. This is the case when the oracle answers with the two-round Feistel network. If not, the adversary knows, that the oracle answers with a random string.
   So he can distinguish between the two-round Feistel network and a random permutation with non-negligible probability, so a two-round Feistel network is not a pseudorandom permutation.

2. **To show:** Prove that a three-round Feistel network using pseudorandom round functions is not a strongly pseudorandom permutation.

   Note: Inversion formula stated in lecture note is wrong.

   position of L3 and R3 should be as $Data = (R3||L3)$

   In lecture note, the position of L3 and R3 is opposite i.e., $Data = (L3||R3)$ This convention is wrong.

   So for avoiding confusion, if $Data = (L3||R3)$, then

   $L_0 = R_3 \oplus f_2(L_3 \oplus f_3(R_3))$

   $R_0 = L_3 \oplus f_3(R_3) \oplus f_1(R_3 \oplus f_2(L_3 \oplus f_3(R_3)))$

   Now coming to proof:

   Let us generalize this notation as $D_{k_1,k_2,k_3}(L||R) = (x||y)$ as the inversion formula for third order. And $E_{k_1,k_2,k_3}(L||R) = (x||y)$ as normal feistel network formula.

   Here is a Adversary that disingushes with high probability

   – Query the decryption oracle with two strings of zero bits: $(a||b) \leftarrow D(0||0)$

Solution of Exercisesheet 10

- Query the encryption oracle: $(c||d) \leftarrow E(0||a)$
- Query the decryption oracle again: $(e||f) \leftarrow D(b \oplus d||c)$
- If $e = c \oplus a$ then return 1, else return 0

Explanation:

- Query the decryption oracle with two strings of zero bits: $(a||b) \leftarrow D(0||0)$
  By using expansion of D
  $a = f_2(f_3(0))$
  $b = f_3(0) \oplus f_1(a)$
- Query the encryption oracle: $(c||d) \leftarrow E(0||a)$ By using expansion of E
  $c = a \oplus f_2(f_1(a))$
  $d = f_1(a) \oplus f_3(c)$
  now on calculating $b \oplus d = f_3(0) \oplus f_3(c)$ we see that $f_1(a)$ cancels out.
- Query the decryption oracle again: $(e||f) \leftarrow D(b \oplus d||c)$ Now again by using the expansion of D we get
  $e = c \oplus a$
- Hence the adversary can check if $e == c \oplus a$ then differentiate if the output is from feistel network or random.

Reference: https://crypto.stackexchange.com/questions/32974/example-of-a-prp-that-is-not-a-strong-prp

Exercise 10-2

To show that computing $\varphi(N)$ is equivalent to factoring N we must prove:

1. If there exists an efficient algorithm to compute $\varphi(N)$ given N, there also exists an efficient algorithm for factoring N.

   $\varphi(N)$ is defined as $|\{a \in \mathbb{N} : 0 \leq a \leq N - 1, gcd(a, N) = 1\}|$
   ($\Rightarrow$ Number of integers relativly prime to N)
   However computing this brute force would require at least N steps which is not efficient. But $\varphi(N)$ can be easily computed with the following formula: $\varphi(N) = \Pi p_i^{e_i-1}(p_i - 1)$. Since $N = pq$ with p and q being primes (per definition of the RSA modulus), $\varphi(N) = (p-1) \cdot (q-1)$. If we can compute $\varphi(N) = x \cdot y$, we also can factorize N as $N = (x + 1) \cdot (y + 1)$.
   $\Rightarrow$ Computing $\varphi(N)$ implies factoring N

2. If there exists an efficient algorithm for factoring N given N, there also exists an efficient algorithm to compute $\varphi(N)$.

   Assuming there is an efficient algorithm for factoring N we can efficiently compute $N = pq$ with p and q being primes (per definition of the RSA modulus).

Solution of Exercisesheet 10

Knowing p and q we can efficiently compute $\varphi(N)$ as $\varphi(N) = (p-1)\cdot(q-1)$ since $\varphi(N) = \Pi p_i^{e_i-1}(p_i - 1)$.
$\Rightarrow$ Factoring N implies computing $\varphi(N)$

Factoring $N \Leftrightarrow$ Computing $\varphi(N)$

Exercise 10-3

F_{k_j} does not lead to a secure strong PRP.
An adversary can query $(0^{\frac{n}{2}} \parallel 0^{\frac{n}{2}})$ and $(0^{\frac{n}{2}} \parallel 1^{\frac{n}{2}})$.
This leads to the L and R values of
$(0^{\frac{n}{2}} \parallel 0^{\frac{n}{2}})$:
$L_0 = 0^{\frac{n}{2}}$ $R_0 = 0^{\frac{n}{2}}$ $L_1 = 0^{\frac{n}{2}}$ $R_1 = k_0$ $L_2 = k_0$ $R_2 = k_0 \oplus k_1$ $L_3 = k_0 \oplus k_1$ $R_3 = k_1 \oplus k_2$ $L_4 = k_1 \oplus k_2$ $R_4 = k_0 \oplus k_2 \oplus k_3$
$(0^{\frac{n}{2}} \parallel 1^{\frac{n}{2}})$:
$L_0 = 0^{\frac{n}{2}}$ $R_0 = 1^{\frac{n}{2}}$ $L_1 = 1^{\frac{n}{2}}$ $R_1 = k_0 \oplus 1^{\frac{n}{2}}$ $L_2 = k_0 \oplus 1^{\frac{n}{2}}$ $R_2 = k_0 \oplus k_1$ $L_3 = k_0 \oplus k_1$ $R_3 = k_1 \oplus k_2 \oplus 1^{\frac{n}{2}}$ $L_4 = k_1 \oplus k_2 \oplus 1^{\frac{n}{2}}$ $R_4 = k_0 \oplus k_2 \oplus k_3 \oplus 1^{\frac{n}{2}}$
When we compare the queries, either L_i, R_i or both from one query is the inverse to L_i, R_i or both, respectively, from the other query.
An adversary can therefore distinguish this construction from a truely random one.