Introduction to Cryptography - Exercise session 4

Prof. Sebastian Faust

November 14, 2018

The purpose of this exercise session is to first exercise (again) the concept of a pseudurorandom function (PRF) and CPA-security. In the second part of the exercise session, we discuss the definition of a pseudorandom permutation (PRP) which was intuitively explained at the end of the lecture. In addition, we explain the concept of Feistel Networks.

Exercise 1 (Extending the range of a PRF)

Let F be a PRF. Below there are two attempts to make another PRF F'. In each case either prove that the result is also a PRF or design a ppt algorithm which breaks it.

(a) $F'_s(x) := F_{0^n}(x) \parallel F_s(x)$, for $F_s\colon \{0,1\}^n \to \{0,1\}^n$.

(b) $F'_s(x) := F_s(0 \parallel x) \parallel F_s(1 \parallel x)$, for $F_s\colon \{0,1\}^{n+1} \to \{0,1\}^n$.

Here "$\|$" denotes concatenation of bit strings.

PSEUDORANDOM PERMUTATION

Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed permutation. F is a *pseudorandom permutation* if for all probabilistic polynomial-time distinguishers D, there exists a negligible function negl such that:

$$|\Pr[\mathsf{D}^{F_k(\cdot)}(1^n) = 1] - \Pr[\mathsf{D}^{f(\cdot)}(1^n) = 1]| \leq \mathsf{negl}(n)$$

where the first probability is taken over uniform choice of $k \in \{0,1\}^n$ and the randomness of D, and the second probability is taken over uniform choice of $f \in \mathsf{Perm}_n$ and the randomness of D.

Exercise 2 (PRP)

Let n be an even number and assume that $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a PRP. We define a fixed-length encryption scheme $\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows: On input $m \in \{0,1\}^{n/2}$ and key $k \in \{0,1\}^n$, algorithm Enc chooses a uniform string $r \in \{0,1\}^{n/2}$ and computes $c := F_k(r||m)$.

(a) Show how the algorithm Dec works.

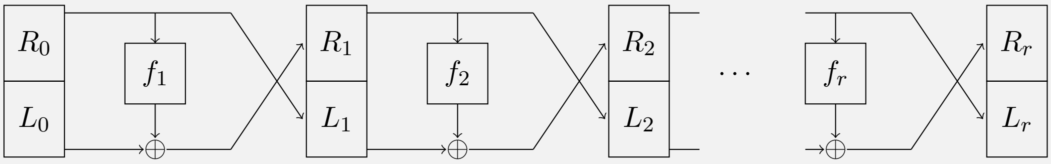
(b) Prove that this scheme is CPA-secure for messages of length $n/2$.

FEISTEL NETWORKS

As discussed during the lecture, Feistel networks offer another approach for constructing block cipher. A Feistel network operates in r rounds. The input $m \in \{0,1\}^\ell$ is split in two halves, i.e. $L_0||R_0 := m$, where $L_0 \in \{0,1\}^{\ell/2}$ is called the left half and $R_0 \in \{0,1\}^{\ell/2}$ is called the right half of the input. In each round $i \in \{1,\ldots,r\}$, a keyed round function $f_i\colon \{0,1\}^{\ell/2} \to \{0,1\}^{\ell/2}$ is applied in the following manner:

$$L_i := R_{i-1} \in \{0,1\}^{\ell/2}$$
$$R_i := L_{i-1} \oplus f_i(R_{i-1}) \in \{0,1\}^{\ell/2}.$$

The output of the r rounds Feistel network is $c := L_r||R_r \in \{0,1\}^\ell$. See the figure below for pictorial representation of the Feistel network.



Exercise 3 (Inverting Feistel network)

Assume that you know all the round functions $\{f_i\}_{i\in[r]}$. Show how to invert the Feistel network, i.e. knowing $c = L_r||R_r$, show how to compute $m = L_0||R_0$ (do not make any addition assumptions on the round functions f_i).

FEISTEL NETWORK using PRF

Let $F : \{0,1\}^{\ell/2} \times \{0,1\}^{\ell/2} \to \{0,1\}^{\ell/2}$ be a PRF. We can use this function to construct a r-round Feistel network in the following way:

1. Choose $(k_1,\ldots,k_r) \leftarrow_\$ \{0,1\}^{r\times\ell/2}$
2. Define $f_i := F_{k_i}$

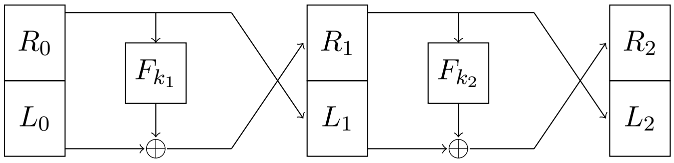
Theorem 1 *For $r \geq 3$, the r-round Feistel network constructed using the PRF F as described above is a PRP.*

In one of the homework exercises, we show that this is not true for $r = 2$.

Voluntary homework exercises

Exercise 4 (Two round Feistel network - Voluntary homework 1)

Let $F : \{0,1\}^{\ell/2} \times \{0,1\}^{\ell/2} \to \{0,1\}^{\ell/2}$ be a PRF. Let us denote $F'\colon \{0,1\}^\ell \to \{0,1\}^\ell$ the 2-round Feistel network constructed using F. Show that F' is **not** a PRP.



Exercise 5 (PRG from PRF - Voluntary homework 2)

Prove that if $F\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a length-preserving PRF, then

$$G(s) := F_s(1)||F_s(2)||\ldots||F_s(l)$$

is a PRG with expansion factor $l \cdot n$.