CS409M: Introduction to Cryptography (Autumn 2024) September 13, 2024

Chalk and Talk-6 Report

Pseudorandom Functions imply Pseudorandom Permutations (Feistel Network)

Instructor:
Sruthi Sekar
sruthi@iitb.ac.in

Submitted by:
Parth Pundalik Pai, Parijat Madras
parthpai@iitb.ac.in, 22B2169@iitb.ac.in

1 Brief overview of the talk

We begin with the discussion of Pseudorandom Functions and how it is different from Pseudorandom Permutations. We look at the Feistel Network construction and its condition for being a Pseudorandom Permutation. Later we look at the Luby-Rackoff formulation and analyse the same for proving our claim of implication between PRFs and PRPs. We introduce a couple of lemmas to assist the proof. We use the hybrid argument to prove one of the lemmas coupled with triangle inequalities etc.

2 Pseudorandom Functions and Pseudorandom Permutations

Definition 1: A Pseudorandom Function F is an efficient function:

$$F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n,$$

such that every efficient algorithm A cannot distinguish well $F_K(\cdot)$ from $R(\cdot)$, for a randomly chosen key $K \in \{0,1\}^k$ and a random function $R : \{0,1\}^n \to \{0,1\}^n$. Intutively it suggests us that we want $A^{F_K(\cdot)}$ to behave like $A^{R(\cdot)}$ for the algorithm A.

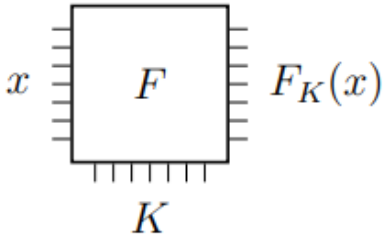

Figure 1: Pseudorandom Function

Definition 2: A Pseudorandom Permutation P is an efficient function:

$$P : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n,$$

such that for every key K, the function P_K mapping $x \mapsto P_K(x)$ is a bijection. In addition to that, we assume that given K, the mapping $x \mapsto P_K(x)$ is efficiently invertible. That is P_K^{-1} is efficient. The security of P states that every efficient algorithm A cannot distinguish well $\langle P_K(\cdot), P_K^{-1}(\cdot)\rangle$ from $\langle \Pi(\cdot), \Pi^{-1}(\cdot)\rangle$, for a randomly chosen key $K \in \{0,1\}^k$

and a random permutation $\Pi : \{0,1\}^n \to \{0,1\}^n$. Intutively, here it suggests us that we want $A^{P_K(\cdot),P_K^{-1}(\cdot)}$ to behave like $A^{\Pi(\cdot),\Pi^{-1}(\cdot)}$ for the algorithm A.

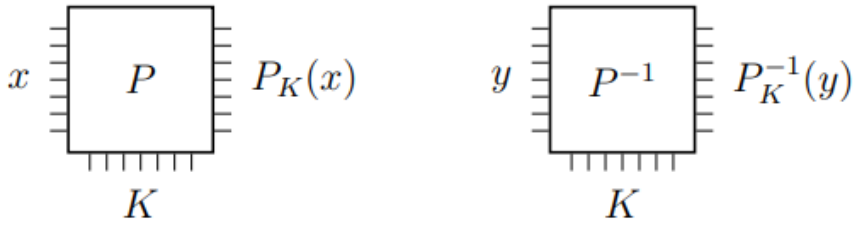

Figure 2: Pseudorandom Permutation

3 Feistel Permutation & Luby-Rackoff Construction

3.1 Feistel Permutation

Given *any* function $F : \{0,1\}^m \to \{0,1\}^m$, we can construct a permutation D_F using a technique named after Horst Feistel, called as Feistel Permutation.

Definiton 3: The definition of $D_F : \{0,1\}^{2m} \to \{0,1\}^{2m}$ is given by

$$D_F(x,y) := (y, F(y) \oplus x), \tag{1}$$

where x and y are m-bit strings. Note that this is an injective function. It is bijective as well since the inverse is the following.

$$D_F^{-1}(z,w) := (F(z) \oplus w, z). \tag{2}$$

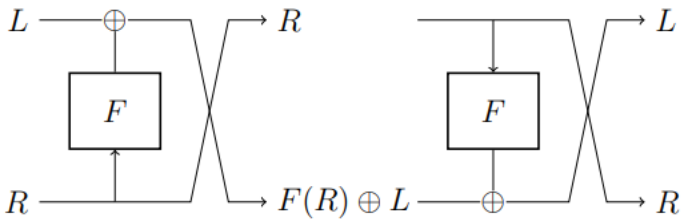

Figure 3: Feistel Permutation

Note that D_F and D_F^{-1} are efficiently computable given F. D_F needs not be a pseudorandom permutation even if F is a pseudorandom function, because the output of $D_F(x,y)$ must contain y, and it is very unlikely that a random permutation will contain entirity of y in its output.

The next intutive step might be to consider applying the Feistel permutation twice. That is, we pick two independent random keys K_1 and K_2, and compose the permutations $P(\cdot) := D_{F_{K_2}}(D_{F_{K_1}}(\cdot))$. But this construction is still insecure, explained by the following example.

Let, $\overline{0}$ denotes the all-zero string of length m, $\overline{1}$ denotes the all-one string of length m, and $F(0^m) = F_{K_1}(0^m)$. This shows that, restricting to the first half, $P(00)$ is the complement of $P(10)$, regardless of F. Similarly its a food for thought why we do not get a pseudorandom permutation if the Feistel network is applied thrice.

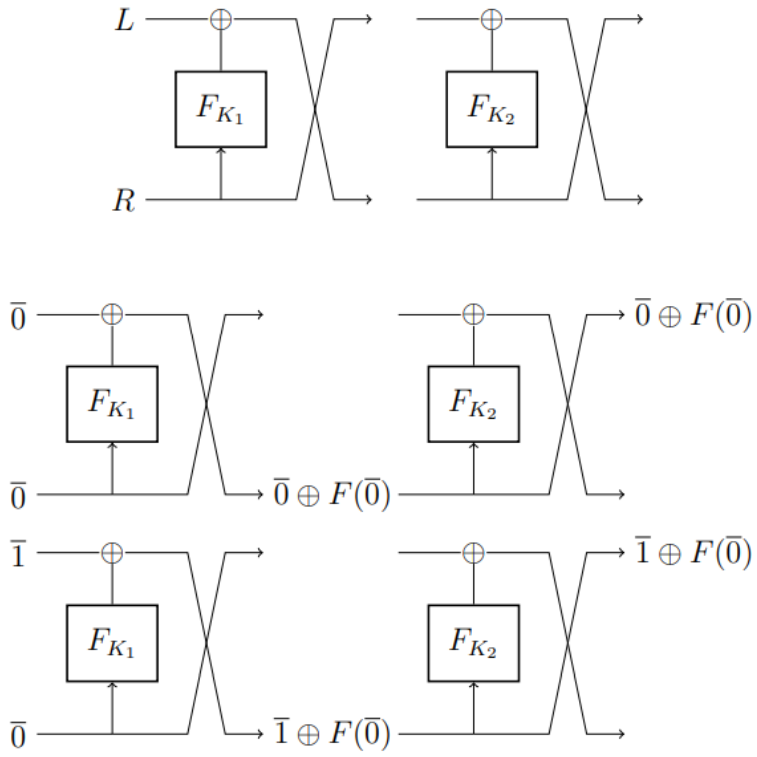

Figure 4: Feistel Not secure

3.2 Luby-Rackoff Construction

Definition 4: Let $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^m$ be a pseudorandom function. We define the following function $P : \{0,1\}^{4k} \times \{0,1\}^{2m} \to \{0,1\}^{2m}$: given a key $K = \langle K_1, \ldots, K_4 \rangle$ and an input x,

$$P_K(x) := D_{F_{K_4}}(D_{F_{K_3}}(D_{F_{K_2}}(D_{F_{K_1}}(x)))). \tag{3}$$

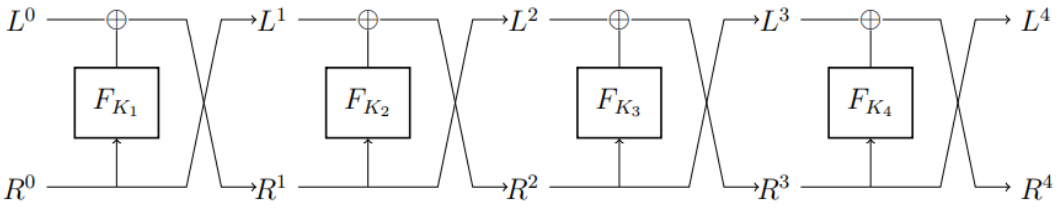

Figure 5: Luby-Rackoff Construction

Note that this is just composing the Feistel network 4 times where the function considered is a Pseudorandom Function. Here the inverse of the following transformation can be found by composing the inverses of the Feistel Permutations.

Similarly, given four random functions $\overline{R} = \langle R_1, \ldots, R_4 \rangle$, $R_i : \{0,1\}^m \to \{0,1\}^m$, let $P_{\overline{R}}$ be the analog of Construction (3) using the random function R_i instead of the pseudorandom functions F_{K_i} in the Feistel network,

$$P_{\overline{R}}(x) = D_{R_4}(D_{R_3}(D_{R_2}(D_{R_1}(x)))) \tag{4}$$

We will need this construction for the proof of the claim using hybrid argument.

4 Proof of the claim: PRF implies PRP

We would like to prove the following theorem using all the tools we have learnt till now.

Theorem 1 (Pseudorandom Permutations from Pseudorandom Functions) If F is a $(O(tr), \epsilon)$-secure pseudorandom function computable in time r, then P is a

$$(t, 4\epsilon + t^2 \cdot 2^{-m} + t^2 \cdot 2^{-2m})$$

secure pseudorandom permutation.

To prove this theorem, we will go with the following approach.

1. P_K is indistinguishable from $P_{\overline{R}}$ or else we can break the pseudorandom function
2. $P_{\overline{R}}$ is indistinguishable from a random permutation

4.1 P_K is indistinguishable from $P_{\overline{R}}$

For proving this first part, we will need the follwing lemma.

Lemma 2 If F is a $(O(tr), \epsilon)$-secure pseudorandom function computable in time r, then for every algorithm A of complexity $\leq t$ we have

$$\left| P_K \left[A^{P_K, P_K^{-1}}() = 1 \right] - P_R \left[A^{P_R, P_R^{-1}}() = 1 \right] \right| \leq 4\epsilon \tag{5}$$

Proof: We will be using Hybrid argument to solve the following. Consider the following five hybrids from $\{0,1\}^{2m} \to \{0,1\}^{2m}$:

- $H_0(\cdot)$: pick random keys K_1, K_2, K_3, K_4,

$$H_0(\cdot) := D_{F_{K_4}}(D_{F_{K_3}}(D_{F_{K_2}}(D_{F_{K_1}}(\cdot)))),$$

- $H_1(\cdot)$: pick random keys K_2, K_3, K_4 and a random function $F_1 : \{0,1\}^m \to \{0,1\}^m$,

$$H_1(\cdot) := D_{F_{K_4}}(D_{F_{K_3}}(D_{F_{K_2}}(D_{F_1}(\cdot)))),$$

- $H_2(\cdot)$: pick random keys K_3, K_4 and random functions $F_1, F_2 : \{0,1\}^m \to \{0,1\}^m$,

$$H_2(\cdot) := D_{F_{K_4}}(D_{F_{K_3}}(D_{F_2}(D_{F_1}(\cdot)))),$$

- $H_3(\cdot)$: pick a random key K_4 and random functions $F_1, F_2, F_3 : \{0,1\}^m \to \{0,1\}^m$,

$$H_3(\cdot) := D_{F_{K_4}}(D_{F_3}(D_{F_2}(D_{F_1}(\cdot)))),$$

- $H_4(\cdot)$: pick random functions $F_1, F_2, F_3, F_4 : \{0,1\}^m \to \{0,1\}^m$,

$$H_4(\cdot) := D_{F_4}(D_{F_3}(D_{F_2}(D_{F_1}(\cdot))))$$

Clearly, referring to (5), H_0 is the first probability of using all pseudorandom functions in the construction, and H_4 is the second probability of using all completely random functions. By triangle inequality, we know that

$$\exists i \quad \left| P\left[A^{H_i, H_i^{-1}} = 1\right] - P\left[A^{H_{i+1}, H_{i+1}^{-1}} = 1\right] \right| > \epsilon. \tag{6}$$

We now construct an algorithm $A^{G(\cdot)}$ of complexity $O(tr)$ that distinguishes whether the oracle $G(\cdot)$ is a pseudorandom function $F_K(\cdot)$ or a random function $R(\cdot)$.
The following are the steps of the algorithm.

- The algorithm A' picks i keys $K_1, K_2, \ldots, K_i$ and initializes $4-i-1$ data structures $S_{i+2}, \ldots, S_4$ to $\emptyset$ to store pairs.
- The algorithm A' simulates $A^{O, O^{-1}}$. Whenever A queries O (or O^{-1}), the simulating algorithm A' uses the four compositions of Feistel permutations as given above in the Luby-Rackoff construction, where
  - On the first i layers, we run the pseudorandom function F using the i keys $K_1, K_2, \ldots, K_i$;
  - On the i-th layer, run the oracle G;
  - On the remaining $4 - i - 1$ layers, simulate a random function.

  In the third case, when a new value for x is needed, use fresh randomness to generate the random function at x and store the key-value pair into the appropriate data structure; otherwise, simply return the value stored in the data structure.

  So basically here we are individually considering the effect of replacing a Pseudorandom function with a completely random function in the Feistel Network construction.

When G is F_K, the algorithm A^G behaves like $A^{H_i, H_i^{-1}}$; when G is a random function R, the algorithm A^G behaves like $A^{H_{i+1}, H_{i+1}^{-1}}$. Rewriting (6),

$$\left| P_K\left[A^{F_K(\cdot)} = 1\right] - P_R\left[A^{R(\cdot)} = 1\right] \right| > \epsilon,$$

and this implies that F is not $(O(tr), \epsilon)$-secure.
Since this contradicts our assumption of F being $(O(tr), \epsilon)$-secure, we have the following result with us by combining the triangle inequality for four hybrids.

$$\left| P_K\left[A^{P_K, P_K^{-1}}() = 1\right] - P_R\left[A^{P_R, P_R^{-1}}() = 1\right] \right| \leq 4\epsilon$$

4.2 $P_{\overline{R}}$ is indistinguishable from a random permutation

Let's look at Lemma 3 for proving this part
Lemma 3 For every algorithm A of complexity $\leq t$, we have

$$\left| P_R\left[A^{P_R, P_R^{-1}}() = 1\right] - P_\Pi\left[A^{\Pi, \Pi^{-1}}() = 1\right] \right| \leq \frac{t^2}{2^{2m}} + \frac{t^2}{2^m},$$

where $\Pi : \{0,1\}^{2m} \to \{0,1\}^{2m}$ is a random permutation.

To prove this, we will apply triangle inequality on Lemma-4 and Lemma-5. A small explaination word here, we say an algorithm A is non-repeating if it never makes an oracle query to which it knows the answer. We shall prove Lemma-3 for non-repeating algorithms. We can extend this proof to any arbitrary algorithm by simulating it using a non-repeating algorithm with comparable complexity such that the algorithm and the simulation have the same output given any oracle permutation.

Lemma 4 Let A be a non-repeating algorithm of complexity at most t. Then

$$\left| P\left[S(A) = 1\right] - P_{\Pi}\left[A^{\Pi,\Pi^{-1}}() = 1\right] \right| \leq \frac{t^2}{2 \cdot 2^{2m}} \tag{7}$$

where $\Pi : \{0,1\}^{2m} \to \{0,1\}^{2m}$ is a random permutation.

Proof of Lemma 4:

$$\left| P\left[S(A) = 1\right] - P_{\Pi}\left[A^{\Pi,\Pi^{-1}}() = 1\right] \right|$$

$$\leq P[\text{when simulating } S, \text{ get answers inconsistent with any permutation}]$$

$$\leq \frac{1}{2^{2m}}(1 + 2 + \cdots + t - 1)$$

$$= \binom{t}{2}\frac{1}{2^{2m}}$$

$$\leq \frac{t^2}{2 \cdot 2^{2m}}.$$

Note that here we applied the idea of collision frequency $coll(q, N)$ (equivalently Birthday paradox) for proving the same.

Lemma 5 For every non-repeating algorithm A of complexity $\leq t$, we have

$$\left| P_R\left[A^{P_R,P_R^{-1}}() = 1\right] - P\left[S(A) = 1\right] \right| \leq \frac{t^2}{2 \cdot 2^{2m}} + \frac{t^2}{2^m}.$$

Proof of Lemma 5:

We define the transcript of A's computation to consist of all the oracle queries made by A. The notation $(x, y, 0)$ represents a query to the π oracle at point x, while $(x, y, 1)$ is a query made to the π^{-1} oracle at y. The set T consists of all valid transcripts for computations where the output of A is 1, while T' is a subset of T containing transcripts which are consistent with π being a permutation.

We write the difference in the probability of A outputting 1 when given oracles (P_R, P_R^{-1}) and when given a random oracle as in $S(A)$ as a sum over transcripts in T:

$$\left| P_F\left[A^{P_R,P_R^{-1}}() = 1\right] - P\left[S(A) = 1\right] \right| = \left| \sum_{\tau \in T} \left(P_F\left[A^{P_R,P_R^{-1}}() \leftarrow \tau\right] - P\left[S(A) \leftarrow \tau\right] \right) \right| \tag{4}$$

We split the sum over T into a sum over T' and $T \setminus T'$ and bound both the terms individually. We first handle the simpler case of the sum over $T \setminus T'$:

$$\left| \sum_{\tau \in T \setminus T'} \left(P_F \left[A^{P_R, P_R^{-1}}() \leftarrow \tau \right] - P\left[S(A) \leftarrow \tau \right] \right) \right| = \sum_{\tau \in T \setminus T'} \left(P\left[S(A) \leftarrow \tau \right] \right) \frac{t^2}{2^{2m}} \tag{5}$$

This relation depicts all those transcripts which are consistent with a permutation. $S(A)$ makes at most t queries to an oracle that answers every query with an independently chosen random string from $\{0,1\}^{2m}$. The probability of having a repetition is at most $\left(\sum_{i=1}^{t-1} i\right)/2^{2m} \leq t^2/2^{2m+1}$.

Another case involves the transcripts in T', which satisfy the criteria of a permutation,yet giving rise to an anomaly. Each x_i can be written as (L_i^0, R_i^0) for strings L_i^0, R_i^0 of length m corresponding to the left and right parts of x_i.Let the number of queries be 'q',bounded by 't'. The output of the construction after iteration k, $0 \leq k \leq 4$, for input x_i is denoted by (L_i^k, R_i^k).

Functions F_1, F_4 are said to be good for the transcript τ if the multisets $\{R_1^1, R_1^2, \ldots, R_1^q\}$ and $\{L_3^1, L_3^2, \ldots, L_3^q\}$ do not contain any repetitions. We bound the probability of F_1 being bad for τ by analyzing what happens when $R_i^1 = R_j^1$ for some i, j:

$$\begin{aligned} R_i^1 &= L_i^0 \oplus F_1(R_i^0) \\ R_j^1 &= L_j^0 \oplus F_1(R_j^0) \\ 0 &= L_i^0 \oplus L_j^0 \oplus F_1(R_i^0) \oplus F_1(R_j^0) \end{aligned} \tag{6}$$

Since we cannot have repeated queries, equation (6) will be true only if $F_1(R_i^0) = s \oplus F_1(R_j^0)$ for a fixed s and pair of distinct strings. This happens with probability $1/2^m$, as the function F_1 takes values from $\{0,1\}^m$ independently and uniformly at random.This gives-

$$P_{F_1}\left[\exists i, j \in [q], R_i^1 = R_j^1 \right] \leq \frac{t^2}{2^{2m+1}} \tag{7}$$

We use a similar argument to bound the probability of F_4 being bad. If $L_i^3 = L_j^3$ for some i, j, we would have:

$$\begin{aligned} L_i^3 &= R_i^4 \oplus F_4(L_i^4) \\ L_j^3 &= R_j^4 \oplus F_4(L_j^4) \\ 0 &= R_i^4 \oplus R_j^4 \oplus F_4(L_i^4) \oplus F_4(L_j^4) \end{aligned} \tag{8}$$

By giving a similar argument we can say,

$$P_{F_4}\left[\exists i, j \in [q], L_i^3 = L_j^3 \right] \leq \frac{t^2}{2^{2m+1}} \tag{9}$$

Equations (7) and (9) together imply that

$$P_{F_1, F_4}\left[F_1, F_4 \text{ not good for transcript } \tau \right] \leq \frac{t^2}{2^m} \tag{10}$$

Since we have proved both Lemma-4 and Lemma-5 here, we can use triangle inequality to obtain the expression for Lemma-3

$$\left| P_R \left[A^{P_R, P_R^{-1}}() = 1 \right] - P_\Pi \left[A^{\Pi, \Pi^{-1}}() = 1 \right] \right| \leq \frac{t^2}{2^{2m}} + \frac{t^2}{2^m},$$

Since we have proved both Lemma-2 and Lemma-3 ,this therefore concludes the proof for Theorem 1

5 Reference Material

- Stanford Crypto Lecture 15 - PRF implies PRP claim
- Stanford Crypto Lecture 16 - Luby-Rackoff Analysis Proof
- Katz and Lindell Book (Second Edition) - Section 6.2.2