ELE2742 / ENG1467 - PUC-Rio, 2019.1

Luby-Rackoff Theorem and related topics

January 10, 2020

Lecturer: Wenfei Wu *Scribe: Xin Lyu and Junkun Chen*

1 Introduction

Feistel Network was a famous design for constructing symmetric ciphers. It was due to German-born physicist and cryptographer Horst Feistel. It was widely deployed in Block ciphers, including the famous Date Encryption Standard (DES). A great advantage of Feistel network is that is has similar encryption and decryption process, hence, the difficuty of implementing this cipher in code or circuit is nearly halved.

After widely application of Feistel Netowrk, people also consider the theoretical analysis of this design. In particulay, Luby and Rackoff [LR88] analyzed the construction, ad proved that if the internal round functions are secure pseudorandom, then 3-rounds of iteration suffices to make the block cipher a pseodorandom permutation, while 4-rounds are sufficient to make it a strong pseodurandom (i.e. under chosen ciphertext attack). Theoretically speaking, Luby and Rackoff's work indicates that pseudorandom function and permutation are equivalent to each other in some sense (that is, under some reasonable assumptions, we have an explicit way to construct PRP/PRF given another).

This project surveys the famous work of Luby and Rackoff. We introduce the general framework to analyze security of pseudo random permutations and functions: the so-called H-coefficient technique. We used this tool to show the security of $3/4$-rounds Feistel networks. We also discuss some recent improvement / analysis of Feistel network.

1.1 Notations

We use $\mathcal{F}_n$ to denote the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$, $\mathcal{P}_n$ to denote the set of all permutations from $\{0,1\}^n$ to itself. Use $x\|y$ to denote concatenation of two strings. Also let $\mathbf{1}[\psi]$ to denote the indicator: i.e. $\mathbf{1}[\psi]$ is 1 if and only if ψ is true statement, otherwise $\mathbf{1}[\psi] = 0$.

1.2 Feistel Network

We give following defintion for Feistel network.

Definition 1.1 (r-round Feistel network)**.** Given $n \geq 1$ be an interger. Let $f_1, \ldots, f_r \in \{0,1\}^n \to \{0,1\}^n$ be r functions. The Feistel network $\Psi(f_1 \ldots f_r, x) : \{0,1\}^n \to \{0,1\}^n$ is defined as follows. For given input $x \in \{0,1\}^{2n}$, write $x = L_0\|R_0$ where $|L_0| = |R_0| = n$. Iteratively define $L_i\|R_i = R_{i-1}\|L_{i-1} \oplus f_i(R_{i-1})$ for $i = 1, \ldots, r$, finally define $\Psi(x) = L_r\|R_r$.

Given this construction, we can design the decryption process accordingly: for given $y \in \{0,1\}^{2n}$, write $y = L_r \| R_r$, the iteratively compute $R_{i-1} = L_i, L_{i-1} = R_i \oplus f_i(L_i)$ for $i = r, \ldots, 1$ and output $L_0 \| R_0$. Observe the decryption process is very similar to encryption. In fact, if we swap the left and right part, then they become two almost identical processes.

For convenience, we will use $N := 2n$ in following presentation to denote the length of input/output strings of the Feistel network.

2 H-coefficient Technique

The definition of H-coefficient captures some common features behind different construction of block ciphers, hence becomes a widely used tool in proving security of pseudorandom functions / permutations. We first give following definition. This proof is borrowed from [NPV17].

Definition 2.1 (H-coefficient)**.** Let $\mathcal{K}$ be a set of r-tuples $(f_1, \ldots, f_r)$ of functions in $\mathbb{F}_n$. Let Ψ be a r-round Feistel network with keyspace $\mathcal{K}$. Let $(a_i)_{i=1}^q$ and $(b_i)_{i=1}^q$ be two sequences of strings of length n. Then the H-coefficient $H(a,b)$ is defiend as the number of keys $(f_1, \ldots, f_r) \in \mathcal{K}$ such that:

$$\Psi(a_i) = b_i, \ \forall 1 \leq i \leq q.$$

To see why H-coefficient is useful, we give following important results.

2.1 Result On CPA

Theorem 2.2. *Let $\alpha, \beta > 0$ be two reals, let E be a subset of $\{0,1\}^{Nq}$ such that $|E| \geq 2^{Nq}(1-\beta)$. If for all sequences $(a_i)_{i=1}^q$ of pairwise distinct elements of $\{0,1\}^n$ and for all sequences $(b_i)_{i=1}^n$ from E, we have:*

$$H \geq \frac{|\mathcal{K}|}{2^{Nq}}(1-\alpha).$$

Then, for every CPA adversary distinguishing the Feistel cipher Ψ from a truly random function, with up to q chosen plaintext queries, we have:

$$\mathrm{adv}\mathbf{CPA} \leq \alpha + \beta.$$

Proof. First of all, we can WLOG assume adversary are deterministic. In fact, any randomized adversary can be seen as a distribution over a set of deterministic one. For a randomized adversary achieving some advantage c, we can always find a deterministic one achieving the same advantage.

Let $\mathcal{A}$ be any adversary which makes q adaptively chosen plaintext queries $(\gamma_i)_{i=1}^q$, let $\delta_i = f(\gamma_i)$ be the responses from challenger. Note that, given the response sequence (δ_i), the query sequence γ_i and adversary's output can be uniquely determined. Hence, we will just write $\mathcal{A}(\delta')$ as the output for function f as long as $\delta(f) = \delta'$, we also write $\gamma(\delta)$ to denote the query patterns (γ_i) given response sequence (δ_i).

Let $p_1 = \mathbf{Pr}[\mathcal{A}(f) = 1 \mid f \xleftarrow{R} \mathcal{F}_N]$ and $p_1' = \mathbf{Pr}[\mathcal{A}(f) = 1 \mid f \xleftarrow{R} \Psi(\mathcal{K})]$. Denote $\Delta = \{\delta \in \{0,1\}^{Nq} : \mathcal{A}(\delta) = 1\}$. We have:

$$\begin{aligned} p_1 &= \frac{1}{|\mathcal{F}_n|} \sum_{f \in \mathcal{F}_n} \mathbf{1}[\mathcal{A}(f) = 1] \\ &= \frac{1}{|\mathcal{F}_n|} \sum_{\delta' \in \Delta} \sum_{f} \mathbf{1}[\delta(f) = \delta'] \\ &= \frac{1}{|\mathcal{F}_n|} \cdot |\Delta| \cdot \frac{|\mathcal{F}_n|}{2^{Nq}} \\ &= \frac{|\Delta|}{2^{Nq}}. \end{aligned}$$

We also have:

$$\begin{aligned} p_1' &= \frac{1}{|\mathcal{K}|} \sum_{(f_1 \ldots f_r)} \mathbf{1}[\mathcal{A}(\Psi(f_i)) = 1] \\ &= \frac{1}{|\mathcal{K}|} \sum_{\delta' \in \Delta} H(\gamma(\delta'), \delta') \\ &\geq \frac{1}{|\mathcal{K}|}(\Delta - \beta 2^{Nq}) \frac{|\mathcal{K}|}{2^{Nq}}(1 - \alpha) \\ &= (1 - \alpha)(p_1 - \beta) \\ &\geq p_1 - \alpha - \beta. \end{aligned}$$

Then we conclude $p_1' - p_1 \leq \alpha + \beta$. We construct another adversary $\mathcal{A}'$ such that $\mathcal{A}'(f) = 1 - \mathcal{A}$, then a same analysis on $\mathcal{A}'$ shows that $p_1 - p_1' \leq \alpha + \beta$, hence, we conclude:

$$\mathrm{adv}\mathbf{CPA} = |p_1 - p_1'| \leq \alpha + \beta.$$

□

Remark 2.3. We give an intuitive understandng of this result. The set E in H-coeffient technique serves as a 'fooling set' in our proof. We showed that is H is large enough, then with high probability the queried patterns lies in E. If it is indeed the case, the adversary is 'fooled': whatever results the challengers give to $\mathcal{A}$, the responses are 'random' enough such that it can not be significantly distinguished from a truly random function.

2.2 Result On CCA

In this section, we will extend our result to CCA security.

Theorem 2.4. *Let $\alpha > 0$ be a real number. If for ALL pairs of sequences $(a_i)_{i=1}^q, (b_i)_{i=1}^q$, where a, b consist of pairwise distinct elements of $\{0,1\}^n$, we have:*

$$H \geq \frac{|\mathcal{K}|}{2^{Nq}}(1 - \alpha).$$

Then, for any CCA adversary D distniguishing between a Feistel networl $\Psi(\mathcal{K})$ and a truly random permutation, with up to q chosen plaintext of chosen ciphertext queries, we have:

$$\mathrm{adv}\mathbf{CCA} \leq \alpha + \frac{q(q-1)}{2^{N+1}}.$$

Proof. Suppose there are q queries made in total. We let γ_i be the sequence of query strings, and δ_i be the sequence of responses. Again, the interaction between adversary and challenger, as well as the output of algorithm are determined by δ. So we use the notation $\mathcal{A}(\delta)$ as above. Let $\Delta = \{\delta \in \{0,1\}^{Nq} : \delta \text{ is 'compatible' with a permutation and } \mathcal{A}(\delta) = 1\}$. By 'compatible' we mean there is indeed some $f \in \mathcal{P}_n$ which induces the sequence (δ_i).

We also define two sequences $(a_i), (b_i)$ for an interaction process, a_i collects the plaintext in each query in order, while b_i collects the ciphertext. Clearly a, b are uniquely determined by (δ_i).

Let $p_1 = \mathbf{Pr}[\mathcal{A}(f) = 1 \mid f \xleftarrow{R} \mathcal{P}_n\}$ and $p_1' = \mathbf{Pr}[\mathcal{A}(f) = 1 \mid f \xleftarrow{R} \Psi(\mathcal{K})\}$. We have following calculations:

$$\begin{aligned}
p_1 &= \frac{1}{|\mathcal{P}_n|} \sum_{f \in \mathcal{P}_n} \mathbf{1}[\mathcal{A}(f) = 1] \\
&= \frac{1}{|\mathcal{P}_n|} \sum_{\delta' \in \Delta} \sum_{f \in \mathcal{P}_n} \mathbf{1}[\delta(f) = \delta'] \\
&= \frac{1}{|\mathcal{P}_n|} \cdot |\Delta| \cdot \frac{|\mathcal{P}_n|}{2^N(2^N-1) \cdot \cdots \cdot (2^N - q + 1)} \\
&= \frac{|\Delta|}{2^{Nq}\left(1 - \frac{q(q-1)}{2^{N+1}}\right)}
\end{aligned}$$

and:

$$\begin{aligned}
p_1' &= \frac{1}{|\mathcal{K}|} \sum_{(f_1 \ldots f_r) \in \mathcal{K}} \mathbf{1}[\mathcal{A}(\Psi(f_i)) = 1] \\
&= \frac{1}{|\mathcal{K}|} \sum_{\delta' \in \Delta} H(a(\delta'), b(\delta')) \\
&\geq \frac{1}{|\mathcal{K}|} \cdot |\Delta| \cdot \frac{|\mathcal{K}|}{2^{Nq}}(1 - \alpha) \\
&\geq p_1 \left(1 - \frac{q(q-1)}{2^{N+1}}\right)(1 - \alpha) \\
&\geq p_1 - \alpha - \frac{q(q-1)}{2^{N+1}}
\end{aligned}$$

It shows $p_1 - p_1' \leq \alpha + \frac{q(q-1)}{2^{N+1}}$. Now, apply the same trick as in above proof, we have:

$$\mathrm{adv}\mathbf{CCA} = |p_1 - p_1'| \leq \alpha + \frac{q(q-1)}{2^{N+1}}$$

as desired. □

3 Luby-Rackoff Theorem

We will use counting arguement to estimate H-coefficient of Ψ. Together with theorems we proved in last section, We can prove Luby-Rackoff theorem.

3.1 3-round Feistel Network

First, we consider the 3-round Feistel network with truly random functions as internal round functions. We have following lowerbound:

Theorem 3.1. *Let Ψ^3 be a Feistel network, with key space $\mathcal{F}_n^3$. Let L_i, R_i, S_i, T_i be elements of $\{0,1\}^n$ such that (L_i, R_i) are pairwise distinct, and S_i are pairwise distinct. Then the H-coefficient between $(L_i \| R_i)_{i=1}^q$ and $(S_i \| T_i)_{i=1}^q$ is bounded by:*

$$H \geq \frac{|\mathcal{F}_n|^3}{2^{Nq}} \left(1 - \frac{q(q-1)}{2^{n+1}}\right).$$

Proof. Write equation for desired f_1, f_2, f_3 as follows:

$$\begin{cases} S_i = R_i \oplus f_2(L_i \oplus f_1(R_i)) \\ T_i = L_i \oplus f_1(R_i) \oplus f_3(S_i) \end{cases}, \forall 1 \leq i \leq q.$$

First, since $L_i \| R_i$ are pairwise distinct, there are at least $|\mathcal{F}_n| \left(1 - \frac{q(q-1)}{2^{n+1}}\right)$ choices of f_1 such that $L_i \oplus f_1(R_i)$ are pairwise distinct. Fix a f_1 with this property, we have $\frac{|\mathcal{F}_n|}{2^{nq}}$ choices of f_2 satisfying first set of equations. Then fix a f_2 with the property, we have $\frac{|\mathcal{F}_n|}{2^{nq}}$ choices of f_3 satisfying second set of equations. Multiply them together, we get the desired lowerbounds. □

Apply this estimation to the general theorem we proved before, we have:

Corollary 3.2. *For any adversary distinguishing Ψ^3 from a truly random function, it holds:*

$$\mathrm{adv}\mathbf{CPA} \leq \frac{q(q-1)}{2^n}$$

Proof. Just apply theorem 2.2 with $E = \{(S_i \| T_i)_{i=1}^q : S_i \text{ are pairwise distinct}\}$, $\alpha = \frac{q(q-1)}{2^{n+1}}$ and $\beta = \frac{q(q-1)}{2^{n+1}}$. □

Obviously, we can replace the internal truly random functions by secure pseudorandom functions, then we derive the security of 3 round Feistel network.

Theorem 3.3. *Let Ψ^3 be a Feistel network with internal key space $\mathcal{K}_1 \times \mathcal{K}_2 \times \mathcal{K}_3$. If $\mathcal{K}_i$ are secure pseudorandom functions, then Ψ^3 is secure pseodorandom permutation against chosen plaintext attack.*

3.2 4-round Feistel Network

4-round Feistel network is secure against chosen ciphertext attack, to prove this, we introduce following estimation on H-coefficients of Ψ^4.

Theorem 3.4. *Let Ψ^4 be a Feistel network, with key space $\mathcal{F}_n^4$. Let L_i, R_i, S_i, T_i be elements of $\{0,1\}^n$ such that $L_i \| R_i$, $S_i \| T_i$ are pairwise distinct respectively. Then the H-coefficient between $(L_i \| R_i)_{i=1}^q$ and $(S_i \| T_i)_{i=1}^q$ is bounded by:*

$$H \geq \frac{|\mathcal{F}_n|^4}{2^{Nq}} \left(1 - \frac{q(q-1)}{2^{n+1}}\right)^2.$$

Proof. Write equations for desired $f_i, i = 1, 2, 3, 4$ as follows:

$$\begin{cases} S_i = L_i \oplus f_1(R_i) \oplus f_3(R_i \oplus f_2(L_i \oplus f_1(R_i))) \\ T_i = R_i \oplus f_2(L_i \oplus f_1(R_i)) \oplus f_4(S_i) \end{cases}, \ \forall 1 \leq i \leq q.$$

First of all, we have at least $|\mathcal{F}_n| \left(1 - \frac{q(q-1)}{2^{n+1}}\right)$ choices of f_1 such that $f_1(R_i) \oplus L_i$ are airwise distinct. Fix one such f_1, we require following conditions for f_2:

- $R_i \oplus f_2(L_i \oplus f_1(R_i))$ are pairwise distinct.
- $T_i \oplus R_i \oplus f_2(L_i \oplus f_1(R_i)) = T_j \oplus R_j \oplus f_2(L_j \oplus f_1(R_j))$, if and only if $S_i = S_j$.

To analyze the condition, let k be the number of independent equalities of the form $S_i = S_j$, i.e. the set $\{S_i\}$ takes $q - k$ distinct values. Note that the second condition fix k positions of f_2 (once after the values $f(x)$ on all necessary points are fixed), and there will never be confliction between these two conditions (since $S_i \| T_i$ are pairwise distinct). Therefore, we have at least $\frac{|\mathcal{F}_n|}{2^{nk}} \left(1 - \frac{q(q-1)}{2^{n+1}}\right)$ choices of f_2. Then fix one such f_2, we have $\frac{|\mathcal{F}_n|}{2^{n(q-k)}}$ choices of f_4 to satisfy second set of equations, as well as $\frac{|\mathcal{F}_n|}{2^{nq}}$ choices of f_3 to satisfy the first set of equations. Multiply them together gives the desired bound. □

Similar to proof of 3-round, we have following deduction immediately:

Corollary 3.5. *For any adversary distinguishing Ψ^4 from a truly random permutation, it holds:*

$$\mathrm{adv}\mathbf{CCA} \leq \frac{q(q-1)}{2^n}.$$

Proof. Apply theorem 2.4 with $\alpha = \frac{q(q-1)}{2^n} - \frac{q^2(q-1)^2}{2^{N+2}}$. □

Theorem 3.6. *Let Ψ^4 be a Feistel network with internal key space $\mathcal{K}_1 \times \mathcal{K}_2 \times \mathcal{K}_3 \times \mathcal{K}_4$. If $\mathcal{K}_i$ are secure pseudorandom functions, then Ψ^4 is secure pseodorandom permutation against chosen ciphertext attack.*

4 Quantum Adversary

4.1 Background

We have shown that any classical algorithm trying to distinguish a Feistel Permutation from a truly random permutation requires at least $\Omega(2^{n/2})$ queries. However, if we do not limit our scope in classical world, things become more interesting and more subtle. There are several examples indicating that quantum algorithms are superior to classical algorithms in solving distinguishing problems, such as the Deutch-Jozza problem, the Bernstein-Vazirani problem, and the Simon problem etc.

In 2010, Kuwakado and Morii [KM10] have shown a quantum adversary for distinguishing a 3-round Feistel permutation with internal permutations and random permutation. Prior to that, Treger and Patarin[PT] have shown that any classical algorithms solving this problem requires $\Theta(2^{n/2})$ time. Given an access to a unitary operator as oracle, their construction runs in $\mathrm{poly}(n)$ time and fails w.p. $O(2^{-n})$. This work has at least following important meanings:

1 Their algorithm is inspired by Simon's algorithm. Indeed, the authors mentioned that it is the first applcition of Simon's problem in hacking a cryptosystem. This story suggests that other well-known quantum algorithms may also be used in solving some natural real-world problems, in which they can outperform any classical algorithms.

2 In 1994, Shor proposed quantum algorithm for solving integer factoring and discrete logarithm [Shor94], which makes a number of classical cryptosystem based on number theory construction become insecure against potential quantum attack. From then on, when people talked about quantum challenge to cryptography, they usually quote Shor's algorithm as example. This work suggests that beyond attacking number theory based public-key cryptosystem, quantum algorithm can also be powerful in attacking other cryptosystem.

In this section, we will review the design and analysis of the algorithm proposed in [KM10], and then talk some of its potential extensions.

4.2 Quantum Attack Game

In [KM10], the quantum algorithm and analysis only worked for the case when internal component functions in Feistel network are permutations. Therefore, we give following definition.

Definition 4.1. Let $n \geq 1$ be fixed, a *3-round Feistel permutation with internal permutations* ($3-\mathrm{FP}^p$ for short) $g : \{0,1\}^{2n} \to \{0,1\}^{2n}$, is defined as follows: For given $x\|y$, $|x| = |y| = n$. Write $x_0 = x, y_0 = y$, let $x_i = y_{i-1}, y_i = x_{i-1} \oplus f_i(y_{i-1})$ for $i = 1, 2, 3$, then let $g(x\|y) = x_3\|y_3$. Here f_1, f_2, f_3 are three truly random permutations.

For the convenience of discussing design and analysis of algorithm, we will give a rigorous definition of attack game here. In quantum setting, the attack game consists of a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. In experiment b ($b \in \{0,1\}$), depending on b, challenger either prepares a $3-\mathrm{FP}^p$

or a truly random permutation (TRP), whichever we denote it by $V : \{0,1\}^{2n} \to \{0,1\}^{2n}$. Then $\mathcal{C}$ provides a query oracle U_V to $\mathcal{A}$, the oracle is defined as:

$$U_V|x\rangle|y\rangle \mapsto |x\rangle|y \oplus F(x)\rangle.$$

The adversary works in a Hilbert space $\mathcal{H}_O \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathcal{H}_R$. Where $\mathcal{H}_O$ is a 1-qubit space to store the output. $\mathcal{H}_X$ and $\mathcal{H}_Y$ are used to send query and receive responses respectively, and $\mathcal{H}_R$ is the working register with unbounded number of qubits. Working in this space, $\mathcal{A}$ can apply any quantum channel (unitary operator, measurements, etc.), or send a query to $\mathcal{C}$, who will then apply the oracle U_V to $\mathcal{H}_X \otimes \mathcal{H}_Y$ and notify $\mathcal{A}$. Finally $\mathcal{A}$ writes her output to $\mathcal{H}_O$ (in classical form, or we can perform a measurement as final step).

4.3 The algorithm

4.3.1 The idea

Before we present the algorithm, we introduce the idea behind its design. Let f be a 3-round Feistel permutation with P_1, P_2, P_3 be its internal permutations. Then for given $xy \in \{0,1\}^{2n}$ where $|x| = |y| = n$, we have:

$$f(x\|y) = y \oplus P_2(x \oplus P_1(y))\|(x \oplus P_1(y)) \oplus P_3(y \oplus P_2(x \oplus P_1(y))),$$

Now, let $\alpha, \beta \in \{0,1\}^n$ be two distinct arbitrary but fixed strings, for given input $x \in \{0,1\}^{2n}$, let $W(x)$ be the first n bits of $f(x)$, i.e. $W(x) = f(x)[0 \dots n-1]$. Consider following function $g : \{0,1\}^{n+1} \to \{0,1\}^n$:

$$g(b\|x) = \begin{cases} W(x\|\alpha) \oplus \alpha, & \text{if } b = 0, \\ W(x\|\beta) \oplus \beta, & \text{otherwise.} \end{cases}$$

Observe that:

$$W(x\|y) \oplus y = P_2(x \oplus P_1(y)).$$

Now, since P_1, P_2 are all permutations, we have an important observation: for $(b,a) \neq (b',a')$, $g(b\|a) = g(b'\|a')$ if and only if $b = b' \oplus 1$ and $a = a' \oplus z$ where $z = P_1(\alpha) \oplus P_1(\beta)$, this is exactly the condition required by Simon's problem.

Definition 4.2 (Simon's Problem)**.** Given a function $f : \{0,1\}^n \to \{0,1\}^n$ and some $s \in \{0,1\}^n$ such that for any $x \neq y$, $f(x) = f(y)$ if and only if $x = y \oplus s$. Simon's problem asks to find the hidden s by an access to an oracle f (the oracle may be quantum).

We claim here that Simon's problem can be sovled in $O(n)$ time (the algorithm and its analysis will be given in next subsection) given an access to quantum oracle. So we can design our adversary as follows:

(a) Given the permutation oracle f, choose some $\alpha, \beta \in \{0,1\}^n$ arbitrarily, construct the quantum gate W as defined above. Then use Simon's algirthm to find the hidden $z = P_1(\alpha) \oplus P_1(\beta)$.

(b) Given z, if f is indeed a Feistel Permutation, then we can predict the output of $W(0\|x)$ given $W(1\|x \oplus z)$, we can use $O(1)$ queries to see if our guess is correct, and adversary outputs 1 if and only if the guess is correct. If f is indeed a Feistel permutation, we make correct prediction w.p. 1, otherwise, we make correct prediction w.p. at most $O(2^{-n})$, so we get an advantage close to 1 in total.

In next several subsections, we will resolve several important issues in the algorithm, and complete its analysis.

4.3.2 Simon's Algorithm

We have defined Simon's problem above, here shows the algorithm:

(a) Prepare the initial state $|0\rangle^{\otimes n}$.

(b) Apply Hadamard gate H to n qubits, we get $|s\rangle = \frac{1}{2^{n/2}} \sum_i |i\rangle$.

(c) Apply the quantum access f: $\frac{1}{2^{n/2}} \sum_i |i\rangle$, then measure the output register (in computational basis): we get $\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$ for some x (this is due to $x \oplus s$ is the only string y such that $f(x) = f(y)$).

(d) Apply Hadamard gate to n qubits again, we get:

$$c \cdot \sum_i \left((-1)^{x \cdot i} + (-1)^{(x+s) \cdot i}\right) |i\rangle = c' \cdot \sum_{i \cdot s = 0} |i\rangle,$$

where c' is some normalization factor.

(e) Measure the state in computational basis, we get some i' such that $(s \cdot i') \equiv 0 \bmod 2$.

(f) Repeat step (a) to (e), and add the result i' in each turn to a set $\mathcal{S}$, until s can be uniquely determined by elements in $\mathcal{S}$ (i.e. $\mathrm{rank}(S) = n - 1$).

We have shown the algorithm, and the analysis is also given in algorithm steps. In each turn, $\mathrm{rank}(S)$ increases w.p. at least $\frac{1}{2}$, and each turn only requires a single access to quantum oracle, hence in expectation the algorithms queries oracle $O(n)$ times and runs in $\mathrm{poly}(n)$ time.

4.3.3 Other issues

Here we introduce two another important issues (which is not mentioned in [KM10] original paper).

The first one is, we want to design the oracle gate W from f. given an access to oracle $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$, we just want it to affect the first n qubits in output regiter. If we simply drop the last n qubits in output register, then we will lose some entanglement between input and output, and the algorithm may not work then.

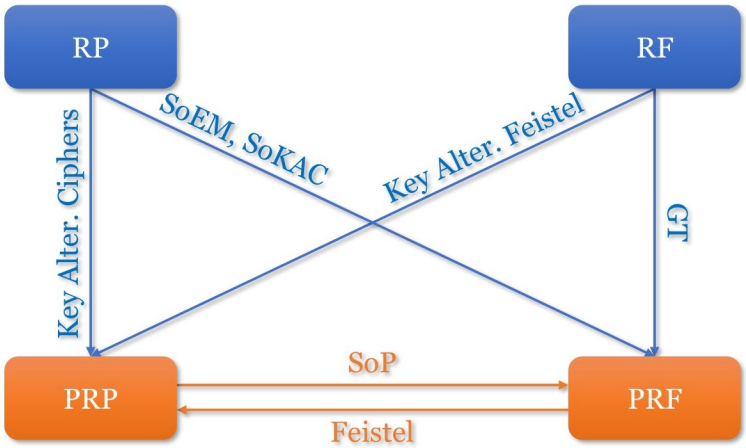
Then solution is rather simple: we use a trick called 'uncomputing'. Write $f(x) = A(x)\|B(x)$ where $|A(x)| = |B(x)| = n$. Given oracle to f, we want to design oracle to A. Here shows how it works: initially we make the second register to be a clean state $|0\rangle^{\oplus 2n}$. Then we just apply f, and then use C-NOT gate to 'copy' the first n-qubits in output register, and then apply f again to 'uncomputing'. Formally, the gate can be described as:

$$|x\rangle|0\rangle|z\rangle \mapsto |x\rangle|f(x)\rangle|z\rangle \mapsto |x\rangle|f(x)\rangle|z \oplus A(x)\rangle \mapsto |x\rangle|0\rangle|z \oplus A(x)\rangle.$$

Another issue is, if f is indeed an FP, we can guarantee the Simon's algorithm termintes in expectation $O(n)$ time. However, if it is not the case, we don't have any guarantee that the algorithm should terminte in the same bound. The first solution to the problem is give a careful analysis of the behavior of Simon's algorithm on a general random permutation. However, we have another simple solution also: we can set a threshold large enough (say $M = 100n$), and in the process of Simon's algorithm, we only permit it to run M turns, if it does not terminate. We just decide the function f as a truly random permutation. Then we derive an algorithm runs in deterministic $O(n)$ time. A simple analysis for the efficiency of Simon's algorithm could show that the modification does not hurt the advantage too much.

5 Conversions between PRP, PRF, RP and RF: An Overview

Conversions between PRP, PRF, (public) RP and (public) RF is a fundamental problem in symmetric key cryptography. For example, Feistel network is a construction of PRF-to-PRP conversion, and there are also conversions in other directions. Nowadays, all possible conversions are constructed by researchers. In this section, we give an overview about researches on conversions.



- **RP $\to$ PRP: Key Alternating Ciphers[1]** The paper uses key alternating cipher

$$E(k_0 \cdots k_t, x) = \pi_t(\cdots \pi_2(\pi_1(x \oplus k_0) \oplus k_1) \oplus k_2 \cdots) \oplus k_t$$

It proved that the construction is secure up to CPA attack complexity $O(2^{tn/(t+1)})$.

- **RP → PRF: SoEM[2]** The paper uses Sum of Even Mansour construction

$$E(K_1, K_2, M) = \pi_1(M \oplus K_1) \oplus K_1 \oplus \pi_2(M \oplus K_2) \oplus K_2$$

that π_1, π_2 are public random permutations. It also proved that if $\pi_1 \neq \pi_2, K_1 \neq K_2$, this construction is secure up to CPA attack complexity $O(2^{2n/3})$.

- **RF → PRP: Key Alternating Feistel[3]** This paper combines key alternating cipher and Feistel networks. It modifies the feistel network that

$$L_i \| R_i \leftarrow R_{i-1} \| L_{i-1} \oplus f_i(R_{i-1} \oplus k_i)$$

It is proved to be secure up to CPA attack complexity $O(2^{tn/(t+1)})$ where $t = \lfloor r/3 \rfloor$ with an r-round Feistel.

- **RF → PRF: GT[4]** This paper gives a construction

$$E(k_1 \cdots k_l, x) = F(x \oplus k_1) \oplus \cdots \oplus F(x \oplus k_l)$$

It is proved that if $k_1, \cdots, k_l$ are independent, the adversary of PRF is at most $q_C(q_C + q_P)^l / 2^{ln}$.

- **PRP → PRF: SoP[5]** This paper designs the backward of Luby-Rackoff by this:

$$F(k_1 \cdots k_t, x) = E((E(k_1, x \gg d) \| \cdots \| E(k_t, x \gg d))_{1,\cdots,l}, x)$$

where $t = \lceil l/n \rceil$.

- **PRF → PRP: Feistel** Introduced in former sections.

References

LR88 Michael Luby and Charles Rackoff. "How to Construct Pseudorandom Permutations from Pseudorandom Functions". In: SIAM Journal on Computing 17.2 (1988), pp. 373–386.

NPV17 Valerie Nachef, Jacques Patarin, and Emmanuel Volte. Feistel Ciphers. Security Proofs and Cryptanalysis. Springer, 2017.

KM10 Kuwakado, Hidenori , and M. Morii . "Quantum Distinguisher Between the 3-Round Feistel Cipher and the Random Permutation." IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings IEEE, 2010.

[1] Bogdanov, A., Knudsen, L.R., Leander, G., Standaert, F., Steinberger, J.P., Tischhauser, E.: Key-Alternating Ciphers in a Provable Setting: Encryption Using a Small Number of Public Permutations - (Extended Abstract). In: Pointcheval and Johansson [59], pp. 45–62

[2] Chen Y.L., Lambooij E., Mennink B. (2019) How to Build Pseudorandom Functions from Public Random Permutations. In: Boldyreva A., Micciancio D. (eds) Advances in Cryptology – CRYPTO 2019. CRYPTO 2019. Lecture Notes in Computer Science, vol 11692. Springer, Cham

[3] Lampe, R., Seurin, Y.: Security Analysis of Key-Alternating Feistel Ciphers. In: Cid and Rechberger [22], pp. 243–264

[4] Gazi, P., Tessaro, S.: Secret-key cryptography from ideal primitives: A systematic overview. In: 2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015. pp. 1–5 (2015)

[5] Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 266–280. Springer (1998)