

One Way Functions and Hard Core Predicates

Mainack Mondal

February 19, 2009

1 Introduction

In this chapter we shall study One Way functions (OWF s) and hard core Predicates (HP s), their formal definitions and some properties. Some concepts and definitions from the previous chapters are extensively used here and so they are defined below for recapitulation.

1.1 Adversary

We model the attacker or adversary as a probabilistic polynomial time algorithm or PPT.

1.2 Concept of Security

Informally an encryption scheme is secure if for each adversary A and for every polynomial $p(\cdot)$, there exist a 'N' such that ,

$$\Pr(A \text{ succeeds in the attack}) < \frac{1}{p(n)}, \forall n > N$$

1.3 Semantic Security or (t, ϵ) -SS

\forall distribution X over $\{0, 1\}^n$

\forall partial information $h : \{0, 1\}^n \rightarrow \{0, 1\}^n$

\forall interesting information $f : \{0, 1\}^n \rightarrow \{0, 1\}$

\forall Adversary A with time complexity $t' < t(n)$, $t(n) = \sum t_d n^d$

\exists Simulating algorithm S such that ,

$$\Pr_{\substack{m \leftarrow \{0,1\}^n \\ (p_k, s_k) \leftarrow G_n}} [A(E(m, p_k), p_k, h(m)) = f(m)] \leq \Pr_{m \leftarrow \{0,1\}^n} [S(h(m)) = f(m)] + \epsilon(n)$$

Where $\epsilon(n)$ is a negligible quantity.

then E (\cdot) is called semantically (t, ϵ) secure or $((t, \epsilon) - SS)$

1.4 Message Indistinguishability or $(t, \varepsilon - \text{MI})$

\forall messages $m_0, m_1 \in \{0, 1\}^n$

\forall Adversary A with time complexity $t' < t(n)$, $t(n) = \sum t_d n^d$

$$\Pr_{\substack{i \in \{0,1\} \\ (p_k, s_k) \leftarrow G_n}} [A(E(m_i, p_k), p_k) = i] \leq \frac{1}{2} + \varepsilon(n)$$

Where $\varepsilon(n)$ is a negligible quantity.

then $E(\cdot)$ is called (t, ε) MI secure or $((t, \varepsilon) - \text{MI})$

Another equivalent definition of $(t, 2\varepsilon) - \text{MI}$ will be used frequently

$$\Pr_{\substack{m_0, m_1 \xrightarrow{U} \{0,1\}^n \\ (p_k, s_k) \leftarrow G_n}} [A(E(m_1, p_k), p_k) = a] - \Pr_{\substack{m_0, m_1 \xrightarrow{U} \{0,1\}^n \\ (p_k, s_k) \leftarrow G_n}} [A(E(m_0, p_k), p_k) = a] \leq 2\varepsilon(n)$$

1.5 n

The security parameter, for example the key length.

1.6 $\varepsilon(n)$

A negligible quantity. Formally speaking a sequence $\{\varepsilon_n\}_{n \in \mathbb{N}}$ (respectively a function $\varepsilon(n) : \mathbb{N} \rightarrow \mathbb{R}$) is called negligible in n if for every positive polynomial $p(\cdot)$ and all sufficiently large n ,

$$\varepsilon(n) < \frac{1}{p(n)}$$

1.7 Uniformly Chosen

If a n bit message m is chosen with uniform probability from $\{0,1\}^n$ then m is defined to be uniformly or randomly chosen and represented symbolically as

$$m \xleftarrow{U} \{0, 1\}^n$$

2 One way functions

2.1 One way functions: Motivation

In secure encryption schemes, the legitimate users should be able to easily decipher the messages using some special information, yet it should be infeasible for an adversary who does not have that special piece of information.

Here one should understand that the term ‘infeasible’ or equivalently ‘computationally hard’ is in the average sense of the word, not in the worst case scenario and so we cannot devise ‘secure’ encryption schemes with every NP complete problems which are essentially hard in the worst case but may not be so in the average case.

Hence the existence of secure encryption schemes implies the existence of a PPT algorithm to generate instances with special information such that

- a. It is easy to solve these instances with the special information.
- b. It is hard, on average to solve these instances without the special information.

The foregoing definition gives rise to the concept of one way function which loosely speaking is easy to compute but hard to invert on average.

2.2 One way functions: Definition

The definition given in this section is called the definition of a strong one way function and is the most popular one.

A function $f : \{0,1\}^ \rightarrow \{0,1\}^*$ is called strongly one way if the following two conditions hold :*

1. *Easy to Compute: \exists deterministic polynomial time algorithm A such that on input x , A outputs $f(x)$, i.e.*

$$A(x) = f(x) \tag{1}$$

2. *Hard to invert : \forall PPT algorithm A' , \forall positive polynomial $p(\cdot)$ and \forall sufficiently large n ,*

$$Pr[A'(f(U_n), 1^n) \in f^{-1}(f(U_n))] < \frac{1}{p(n)} \tag{2}$$

Next we present some explanations useful for understanding the definition.

2.2.1 U_n

A random variable which is uniformly distributed over $\{0,1\}^n$. So the probability is taken over all possible values of U_n . We can also denote U_n by an equivalent uniformly distributed random variable X .

2.2.2 $f^{-1}(f(U_n))$

A' need not to output any specific preimage of $f(X)$. Any element $y \in f^{-1}(f(X))$ will be fine. Although if $f(X)$ is bijective then y will be unique.

2.2.3 1^n

In addition to the function value we also give the length of the desired output in unary notation , namely 1^n . The main reason is to set the variable based on which the complexity will be measured. (that is to define the ‘ n ’ in the notation $O(f(n))$).

Otherwise one may drastically shrink the input and claim it will be a one way function just for the reason that any algorithm that computes the preimage runs in exponential time complexity w.r.t. the output size. To stop that we pass the

desired size and define the complexity of the adversary w.r.t. that.

Example:

Consider the function $f(x) = y$, where $x \in \{0, 1\}^n$ and
 $y =$ binary representation of the length of x
 $=$ binary representation of $|x|$
 So, $|f(x)| = \log_2|x|$

Given $f(x)$ one can compute the preimage quite easily.
 Namely, $f^{-1}(f(x)) \ni 0^{|x|} = 0^{2^{|f(x)|}}$
 this trivial construction requires exponential time complexity w.r.t the size
 of $f(x)$ but linear time complexity w.r.t. size of x .
 Thus passing $1^{|x|}$ stops this function to claim itself an OWF. ■

Note in the special case of length preserving functions i.e. where

$$|f(x)| = |x|$$

This particular information is redundant.
 Thus if f in the definition 2.2 is bijective, length preserving and X is a uniformly
 distributed random variable over $\{0, 1\}^n$ then equation (2) reduces to the fact
 that f will be called a strong OWF if \forall PPT algorithm A' and all sufficiently
 large n

$$Pr[A'(f(X)) = X] < \varepsilon(n) \tag{3}$$

$\varepsilon(n)$ is a negligible quantity using formal notions.

3 Trapdoor One Way Permutation

3.1 Trapdoor One Way Permutation: Motivation

From the discussion in the section 2 we can infer that OWF satisfy the require-
 ment of a function which is easy to compute but hard to invert. But to build a
 secure encryption scheme one should also generate the special information that
 will help the legitimate user to invert the function efficiently, otherwise in the
 scheme the legitimate user will also be at the same stand as that of adversary.

This special and secret information known to the legitimate user and not to
 the adversary is called trap door information and based on that we define the
 trap door one way permutations in the next section which effectively touches
 the holy grail of providing a secure encryption scheme.

3.2 Trapdoor One Way Permutation: Informal Definition

The triplet of algorithms (G, F, I) is called a family of trapdoor permutations where,

G generates the pair (k, t_k) which is the key - trapdoor information pair ,

$F(., k)$ is a bijective function

$I(., k, t_k)$ gives the inverse of $F(., k)$

Such that all the G,F,I are deterministic polynomial time algorithm

And also \forall PPT algorithm A' , \forall positive polynomial $p(n)$ and a uniformly distributed random variable X over $\{0, 1\}^n$, there exists a N such that

$$Pr[A'(f(X), k) = X] < \frac{1}{p(n)}, \forall n > N \quad (4)$$

The term permutation comes from the fact that F is bijective and hence F is nothing but a permutation of inputs.

3.3 An Example of Trapdoor One Way Permutation: RSA

In RSA,

$$F : f(x) = x^e \text{ mod } pq$$

p, q are odd primes.

We know efficient deterministic $O((\log pq)^3)$ algorithms to compute $f(x)$ from x but there exist no known efficient method to compute x given $f(x)$, e , and $N (= pq)$

Now if the information

$$d = e^{-1} \text{ mod } ((p-1)(q-1))$$

is given its easy to compute x from $f(x)$.

We can model this decryption algorithm as I

Hence RSA is a strong candidate for trap door one way permutation.

4 Hard Core Predicates

4.1 Hard Core Predicates: Motivation

Saying that a function f is one way implies that given $f(x)$ it is hard to extract the preimage of $f(x)$, but it may be easy to retrieve some information about x from the knowledge of $f(x)$. So one should not use these 'Unsafe' information for encryption purpose.

Also there may be (or intuitively should be) some information or rather Boolean predicates related to x which are hard to compute from $f(x)$ which makes $f^{-1}(f(x))$ hard to get from $f(x)$.

Informally this hard to get information can be considered as hard core predicate or hp for the OWF f

4.2 Hard Core Predicates: Definition

A polynomial time computable predicate $B : \{0,1\}^n \rightarrow \{0,1\}$ is called a hard core of a function f if \forall PPT algorithm A' , \forall positive polynomial $p(\cdot)$ there exist a N such that $\forall n > N$

$$\Pr[A'(f(X)) = B(X)] \leq \frac{1}{2} + \frac{1}{p(n)} \quad (5)$$

Where X is distributed uniformly over $\{0,1\}^n$

We call the hardcore predicate (t, ε) -hp iff \forall PPT algorithm A' with running time $t' = t(n) = \sum t_{an^d}$, we have

$$\Pr[A'(f(X)) = B(X)] \leq \frac{1}{2} + \varepsilon(n) \quad (6)$$

here $\varepsilon(n)$ is a negligible quantity.

4.3 Hard Core Predicates for Trap Door Permutation

A polynomial time algorithm $B : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}$ is called the hard core of the one way trapdoor permutation (G, F, I) if \forall PPT algorithm A' and \forall positive polynomial $p(\cdot)$, $\exists N$ such that $\forall n > N$

$$\Pr[A'(F(X, k), k) = B(X, k)] \leq \frac{1}{2} + \frac{1}{p(n)} \quad (7)$$

or equivalently ,

$$\Pr[A'(F(X, k), k) = B(X, k)] \leq \frac{1}{2} + \varepsilon(n) \quad (8)$$

here $\varepsilon(n)$ is a negligible quantity.

4.4 Hard Core Predicate: Examples

4.4.1 Example 1

Let $X \xleftarrow{U} \{0,1\}^n$, $X = \sum x_i 2^i$ and Given f is some very good OWF. Now prove or disprove the following statement

Since we can never get any information about x from $f(x)$ hence the following predicate is a hp for any OWF f

$$B(X) = x_1 \oplus x_2 \oplus \dots \oplus x_n$$

Solution:

Let us consider a OWF $f(X)$ for which B is a hp. Now construct another function $g(X)$ as follows

$$g(X) = f(X) \parallel \bigoplus_{i=1}^n x_i$$

this is also a OWF but at the same time B is not a hp for g, since we can just output the last bit of g as the predicate B.
Hence the claim is false. ■

4.4.2 Example 2

Consider the one way trapdoor permutation RSA where

$$F : f(x) = x^e \text{ mod } pq$$

p, q are odd primes, $X \in \{0, 1\}^*$

Claim: The predicate

$$B(x, n) = \left(\left(\frac{x}{n} \right) + 1 \right) \text{ mod } 2$$

is not an hp for RSA where

$$\left(\frac{\cdot}{\cdot} \right) = \text{jacobi symbol}$$

Proof

By property of RSA $n = pq$, p, q , are odd primes
Thus $\varphi(n) = (p-1)(q-1) = \text{even number}$
Since $\text{gcd}(e, \varphi(n)) = 1$
Hence e is odd

$$\begin{aligned} \text{So } \left(\frac{m}{n} \right)^e &= \left(\frac{m}{n} \right), \forall m, n \\ \Rightarrow \left(\frac{y}{n} \right) &= \left(\frac{x}{n} \right) \\ &= \left(\frac{x}{n} \right)^e \\ &= \left(\frac{x}{n} \right) \end{aligned}$$

Consider the algorithm,

```
A'(f(x),n){
  return ( [ ( ( (f(x)/n) + 1) mod 2 ] );
}
```

Thus, $\Pr [A'(f(x),n) = B(x, n)] = 1 > \frac{1}{2} + \epsilon(n) \forall \epsilon(n)$
Hence by definition (8)
B is not a hp for RSA ■

Finding a predicate which is not a hp for RSA except Jacobi symbol is a non trivial research problem.

4.4.3 Example 3

Theorem: the predicate

$$B(x, n) = x \bmod 2, \text{ i.e. lsb of } x$$

And the predicate

$$B'(x, n) = \text{msb of } x$$

Are both hard core predicates for RSA.

Proof

We shall not give a full proof of this claim which can already be found in literature. Instead we make some assumptions and show that based on those if B and B' are not hp for RSA then RSA is not an OWF that is f can be inverted easily without the secret information.

We shall make use of the following lemma and give its proof after this theorem.

$$\text{lsb}(2x \bmod n) = \text{msb}(x \bmod n)$$

here $n = pq$, p, q are odd primes and without loss of generality $x < n$
let B is not an hp, moreover there exist some PPT algorithm A such that

$$\Pr[A(f(x), n) = B(x, n)] = 1 \tag{9}$$

Consider the algorithm,

$$A'(f(x), n) \{ \\ \quad \text{return } [A((f(x) \times 2^e), n)]; \\ \}$$

By the lemma $A(f(2x), n) = B'(x, n)$

$$\begin{aligned} \text{now, } f(2x) &= (2x)^e \bmod n \\ &= 2^e \times f(x) \bmod n \end{aligned}$$

$$\begin{aligned} \text{Hence, } A'(f(x), n) &= A(f(x) \times 2^e, n) \\ &= A(f(2x), n) \\ &= B'(x, n) \end{aligned}$$

Using the fact that $x \xleftarrow{U} \{0, 1\}^n \Rightarrow 2x \xleftarrow{U} \{0, 1\}^n$ and equation (9)

$$Pr[A'(f(x), n) = B'(x, n)] = 1$$

We can easily see that,

$$A'[f(x), n] = 0 \Rightarrow x \in [0, \frac{n}{2})$$

$$A'[f(2x), n] = 0 \Rightarrow x \in [0, \frac{n}{4}) \cup [\frac{n}{2}, \frac{3n}{4})$$

and in general,

$$A'[f(2^i x), n] = 0 \Rightarrow x \in \bigcup_{j=0}^{2^i-1} [\frac{jn}{2^i}, \frac{2j+1}{2^i}n)$$

Hence we can devise a binary search which shall call $A'O(\log n)$ times and compute x from $f(x)$, Let the search algorithm is called A'' .

Then,

$$Pr[A''(f(x), n) = x] = 1 > \varepsilon(n), \quad \forall \varepsilon(n)$$

Which contradicts the fact that RSA is a OWF.

Hence B and B' is hp. ■

Proof of the lemma:

If $0 \leq x < \frac{n}{2}$, $msb(x) = 0$

and then $0 \leq 2x < n$

$$\Rightarrow 2x \bmod n = 2x$$

$$\Rightarrow (2x \bmod n) \bmod 2 = 0$$

$$\Rightarrow lsb(2x) = 0 \text{ for } 0 \leq x < \frac{n}{2}$$

$$\Rightarrow lsb(2x) = msb(x), \text{ for } 0 \leq x < \frac{n}{2}$$

If $\frac{n}{2} \leq x < n$, $msb(x) = 1$

also $0 \leq 2x - n < n - 2$

hence, $2x \bmod n = 2x - n$

now, $n = pq$, p and q are odd, so n is odd

$$\Rightarrow (2x - n) \text{ is odd}$$

$$\Rightarrow (2x - n) \bmod 2 = 1$$

$$\Rightarrow (2x \bmod n) \bmod 2 = 1$$

$$\Rightarrow lsb(2x) = msb(x) \text{ for } \frac{n}{2} \leq x < n$$

combining , for $0 \leq x < n$, $lsb(2x) = msb(x)$ ■

5 Goldwasser Michelle Encryption Scheme

Now we ask the question which is quite natural at this point, why bother about hp? Can we make any use of them? Can we make any encryption scheme with them? The answer is yes and the scheme is a pretty good one.

We first state the encryption scheme to encrypt one bit given a OWF and its hp. We analyse the security of the scheme and extend our results to multiple bits.

5.1 Encryption of one bit using hp

5.1.1 Algorithm

Given is a family of trapdoor permutations (G, F, I) and a hard core predicate $B(X, k)$ for F . Here we want to encrypt a bit b which is the secret information.

SCHEME $((G, F, I), B, b)$ {

*/** key generation */*

1. Generate the pair (k, t_k) using G

*/** Encryption $E_{GM}(b, k)$ */*

1. pick $X \xleftarrow{U} \{0, 1\}^n$
2. return $(F(X, k), b \oplus B(X, k))$

*/** Decryption $D_{GM}(c, F(X, k))$ */*

1. $X = I[F(X, k), t_k]$
2. return $(c \oplus B(X, k))$

}

5.1.2 Example

Consider RSA as the trapdoor one way permutation and $B(X, k) = X \bmod 2$ as the hp for RSA.

Thus for encrypting the bit 'b' the encrypted output is

$$\{X^e \bmod N, b \oplus X \bmod 2\}$$

5.1.3 Security Analysis

Theorem: GM encryption scheme for single bit i.e. E_{GM} is MI secure.

Proof

Suppose the encryption scheme is not (t, ϵ) - MI secure,

So \exists a PPT algorithm A' such that

$$\Pr_{\substack{b \in \{0,1\} \\ (p_k, s_k) \leftarrow G_n \\ X \xleftarrow{U} \{0,1\}^n}} [A(F(X, k), b \oplus B(X, k), k) = b] > \frac{1}{2} + \epsilon(n)$$

Consider the following algorithm A'

```
A'(y, k){
    1. pick random c ∈ {0, 1}
    2. return (c ⊕ A(y, c, k))
}
```

$$\begin{aligned} \text{So, } & \Pr_{X \xleftarrow{U} \{0,1\}^n} [A'(F(X, k), k) = B(X, k)] \\ = & \Pr_{\substack{c \in \{0,1\} \\ (p_k, s_k) \leftarrow G_n \\ X \xleftarrow{U} \{0,1\}^n}} [A(F(X, k), c, k) = B(X, k) \oplus c] > \frac{1}{2} + \epsilon(n) \end{aligned}$$

Since A' is a PPT algorithm just as A .So B is not a hp according to definition.

This is a contradiction.

Hence the primary assumption was wrong.

Hence E_{GM} is MI secure.

Hence proved ■

5.2 Encryption of multiple bits using hp

5.2.1 Algorithm

Given is a family of trapdoor permutations (G, F, I) and a hard core predicate $B(X, k)$ for F .Here we encrypt the n bit message m ($= m[1]m[2]...m[n]$)

SCHEME $((G, F, I), B, m)$ {

*/** key generation **/*

1. Generate the pair (k, t_k) using G

```

                /*** Encryption  $E'_{GM}(m,k)$  ***/
for( i = 1 to n ) {
    1. pick  $X \xleftarrow{U} \{0,1\}^n$ 
    2. return  $(F(X,k), m[i] \oplus B(X,k))$ 
}

                /*** Decryption  $D'_{GM}(d, F(X,k))$  ***/
for( i = 1 to n ) {
    1.  $X = I[F(X,k), t_k]$ 
    2. return  $(d[i] \oplus B(X,k))$ 
}
}

```

5.2.2 Security Analysis

Theorem: GM encryption scheme for multiple bits is MI secure.

Proof

Suppose the multiple bit encryption scheme is not MI secure, then using the alternative definition of $(t, 2\varepsilon)$ - MI from definition 1.4

\exists PPT A and $\exists m, m' \in \{0,1\}^n$ such that

$$\Pr[A(E'_{GM}(m, k), k) = 1] - \Pr[A(E'_{GM}(m', k), k) = 1] > 2\varepsilon(n) \quad (10)$$

consider the following hybrid construction

$$\begin{aligned} \Pr[A(E_{GM}(m[1]))(E_{GM}(m[2])) \dots (E_{GM}(m[n])) = 1] &= p_1 \\ \Pr[A(E_{GM}(m'[1]))(E_{GM}(m[2])) \dots (E_{GM}(m[n])) = 1] &= p_2 \\ \Pr[A(E_{GM}(m'[1]))(E_{GM}(m'[2])) \dots (E_{GM}(m[n])) = 1] &= p_3 \end{aligned}$$

⋮

$$\Pr[A(E_{GM}(m'[1]))(E_{GM}(m'[2])) \dots (E_{GM}(m'[n])) = 1] = p_n$$

where E_{GM} is the single bit GM encryption
and E'_{GM} is the multiple bit GM encryption

Hence from (10)

$$\begin{aligned}
& p_1 - p_n > 2\varepsilon(n) \\
& \Rightarrow \sum_{i=1}^{n-1} (p_i - p_{i+1}) > 2\varepsilon(n) \\
& \Rightarrow \exists l \text{ such that } (p_l - p_{l+1}) > \frac{2\varepsilon(n)}{n}
\end{aligned}$$

which translates to

$$\begin{aligned}
& \Pr[A(E_{GM}(m'[1]))(E_{GM}(m'[2]))\dots \\
& \quad \dots(E_{GM}(m'[l]))(E_{GM}(m[l+1]))\dots(E_{GM}(m[n])) = 1] \\
& \quad - \Pr[A(E_{GM}(m'[1]))(E_{GM}(m'[2]))\dots \\
& \quad \dots\dots\dots(E_{GM}(m'[l]))(E_{GM}(m'[l+1]))\dots(E_{GM}(m[n])) = 1] > \frac{2\varepsilon(n)}{n}
\end{aligned}$$

now consider the algorithm A' ,

```

A'(c, k){
  compute, c1 = EGM(m[1])
           c2 = EGM(m[2])
           .
           .
           cl = EGM(m[l])
           cl+2 = EGM(m[l+2])
           .
           .
           cn = EGM(m[n])
  return A(c1c2c3...clcl+2...cn)
}

```

clearly A' is a PPT algorithm , and

$$\begin{aligned}
& \Pr[A'(E_{GM}(m[l+1]), k) = 1] - \Pr[A'(E_{GM}(m'[l+1]), k) = 1] \\
& = \Pr[A(E_{GM}(m'[1]))(E_{GM}(m'[2]))\dots \\
& \quad \dots(E_{GM}(m'[l]))(E_{GM}(m[l+1]))\dots(E_{GM}(m[n])) = 1] \\
& \quad - \Pr[A(E_{GM}(m'[1]))(E_{GM}(m'[2]))\dots \\
& \quad \dots\dots\dots(E_{GM}(m'[l]))(E_{GM}(m'[l+1]))\dots(E_{GM}(m[n])) = 1] \\
& > \frac{2\varepsilon(n)}{n} \\
& \Rightarrow \Pr[A'(E_{GM}(m[l+1]), k) = 1] - \Pr[A'(E_{GM}(m'[l+1]), k) = 1] > \frac{2\varepsilon(n)}{n}
\end{aligned}$$

Hence by definition 1.4 E_{GM} or single bit encryption is not MI secure which is a direct contradiction to the theorem 5.1.3 which was proved independent of this theorem.

Hence the assumption was wrong and multiple bit GM encryption is MI secure. ■

6 A hp for any one way function: Goldrich Levin theorem

As we have seen in example 4.4.1 that to define a hp given any OWF is not an easy task. But at the same time the natural question that comes to us is there any such predicate exists given an OWF in the first place. The answer is positive and our next and last result of this chapter, the Goldrich Levin theorem of the G-L theorem states just that.

6.1 G-L Theorem

Informal statement

Given (G, F, I) a family of trapdoor permutations, define (G', F', I') , another family of trapdoor permutations as

$$I'((z, r), t_k) = I((z, t_k), r)$$

$$F'((x, r), k) = F((x, k), r)$$

then the inner product of x and r mod 2 i.e.

$$B(x, r) = \bigoplus_{i=1}^n x_i r_i$$

is a hard core predicate for (G, F', I')

Formal statement

Let F be an arbitrary strong one-way function, and let G be defined by $G(x, r) = (F(x), r)$, where $|x| = |r|$. Let $B(x, r)$ denote the inner product mod 2 of the binary vectors x and r . Then the predicate B is a hard-core of the function G .

We would like to point out some interesting aspects of this theorem.

1. The immediate question that comes to anybody what is r and where does it come from ? The answer is that just be content thinking of r as a random string over $\{0, 1\}^n$.

So, the theorem translates to the fact that if F is strongly one-way, then it is infeasible to guess the exclusive-OR (XOR) of a random subset of the bits of x when given $F(x)$ and the subset itself. Because when r is random

$$B(x, r) = \bigoplus_{i=1}^n x_i r_i$$

just represents the xor of random bits of x .

2. We stress that the theorem requires that F be strongly one-way and that the conclusion is false if F is only weakly one-way.
3. G is also strongly one-way. We point out that G maintains other properties of F , such as being length-preserving and being one-to-one.

We prove this theorem using its contrapositive, namely if there is an algorithm that can compute $B(x, r)$ from only the knowledge of $(F(x), r)$ with a high probability, then there will be some algorithm which computes x from the knowledge of $F(x)$ only, with a high probability, i.e. $F(x)$ is not a strong OWF in that case. For providing insight into the proof we provide three stages of the proof, in each stage we lower the probability bound for the algorithm that computes $B(x, r)$ from only the knowledge of $(F(x), r)$. But before we state all these proofs we would like to state and prove some very important results in probability which will be proven really helpful not only in this chapter but also in the coming lessons.

6.2 Important results from probability

6.2.1 Markov Inequality

Let X be a non-negative random variable and v a real number. Then

$$\Pr[X \geq v] \leq \frac{E(X)}{v}$$

Proof

$$\begin{aligned} E(X) &= \sum_x \Pr[X = x] \cdot x \\ &\geq \sum_{x < v} \Pr[X = x] \cdot 0 + \sum_{x \geq v} \Pr[X = x] \cdot v \\ &= \Pr[X \geq v] \cdot v \end{aligned}$$

$$\Rightarrow \Pr[X \geq v] \leq \frac{E(X)}{v} \blacksquare$$

6.2.2 Chebyshev's Inequality

Let X be a random variable and $\delta > 0$, Then

$$\Pr[|X - E(X)| \geq \delta] \leq \frac{Var(X)}{\delta^2}$$

Proof:

Define a new random variable $Y = (X - E(X))^2$ Using Markov's inequality

$$\begin{aligned} \Pr[Y \geq \delta^2] &\leq \frac{E(Y)}{\delta^2} \\ \Rightarrow \Pr[(X - E(X))^2 \geq \delta^2] &\leq \frac{E((X - E(X))^2)}{\delta^2} \\ \Rightarrow \Pr[|X - E(X)| \geq \delta] &\leq \frac{Var(X)}{\delta^2} \text{ Since, } Var(X) = E((X - E(X))^2) \blacksquare \end{aligned}$$

6.2.3 Chernoff Bound

Let X_1, X_2, \dots, X_n be independent 0-1 random variables, so that $\Pr[X_i = 1] = p$ for each i . Then for all ε , $0 < \varepsilon \leq p(1-p)$, and all $p \leq \frac{1}{2}$ we have

$$\Pr \left[\left| \frac{\sum_{i=1}^n X_i}{n} - p \right| > \varepsilon \right] < 2 \cdot e^{-\frac{\varepsilon^2}{2p(1-p)} \cdot n}$$

6.3 Proof of G-L Theorem: Case 1

6.3.1 Statement

If F is an OWF then there is no such PPT algorithm A for which

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] = 1$$

Proof

We have a PPT algorithm A such that

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] = 1$$

We define another algorithm A' as follows

```

A'(F(x)) {
    e1 = 000.....01
    e2 = 000.....10
    .
    en = 100.....00
    for (i = 1 to n)
        return(A(F(x), ei) = B(x, ei))
}

```

observe, $B(x, e_i) = x_i$, hence A' computes all the bits of x also A' is a PPT algorithm since it calls A polynomial times. Hence A' is a PPT algorithm which computes all the bits of x .

$$\Pr[A'(F(x), r) = x] = 1$$

Hence F is not an OWF. This is a contradiction.

So no such A exist which can compute $B(x, r)$ from only the knowledge of $F(x, r)$ and r with such high probability. ■

6.4 Proof of G-L Theorem: Case 2

Now we lower the probability bound from 1 to $(\frac{3}{4} + \varepsilon(n))$ on the PPT algorithm A which can now compute $B(x, r)$ from $F(x)$ and r with a probability more than $(\frac{3}{4} + \varepsilon(n))$. In that case we have to prove the existence of another PPT algorithm A' which can compute x from $F(x)$ only with a high probability. So we are given a PPT algorithm A such that

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] > \frac{3}{4} + \varepsilon(n)$$

Now first we take a look why the strategy that we have followed in the proof 6.1 does not work here, i.e. merely giving n inputs of e_i does not work here. The reasons are :

1. It may be that A was not succeeded to compute $B(x, r)$ when $r = e_i$ since A is probabilistic.
2. The algorithm A' in 6.1 has no means to understand A has succeeded or not in computing x_i

To overcome these difficulties here is the idea

Run A "multiple times" and take the majority.

But there are two things to prove here

1. When you run A multiple times the majority of the results really gives the correct result with a high probability.
2. Multiple times are fine .But what exactly is the order of the no. Of iterations? It should be polynomial to prove our result.

We shall prove these points in this section. That is in this process we really have a majority of results equal to the correct result. The second point will be proven at the section 6.2

6.4.1 Lemma:

$$B(x, r) \oplus B(x, r \oplus e_i) = B(x \oplus e_i) = x_i$$

Proof

$$\begin{aligned} B(x, r) \oplus B(x, r \oplus e_i) &= \bigoplus_{j=1}^n x_j r_j \oplus \bigoplus_{j=1}^n x_j (r_j \oplus e_i) \\ &= \left(\bigoplus_{j=1}^n x_j r_j \oplus \bigoplus_{j=1}^n x_j r_j \right) \oplus \bigoplus_{j=1}^n x_j e_i \\ &= \bigoplus_{j=1}^n x_j e_i = B(x, e_i) = x_i \end{aligned}$$

Hence Proved ■

So what we can do is that calculate both the predicates and from that calculate x_i . Our next two theorems show that for a lots of x 's chosen from the message space i.e. $\{0, 1\}^n$ the probability that A answers both the queries correctly is fairly high.

6.4.2 Theorem:

$$\text{If } \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq p + \varepsilon(n), \quad 0 < p < 1$$

Then \exists a set $S_n \subseteq \{0, 1\}^n$ of size at least $\frac{\varepsilon(n)}{2} 2^n$, where $\forall x \in \{0, 1\}^n$

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq p + \frac{\varepsilon(n)}{2}, \quad p \in (0, 1)$$

Proof

Define

$$S_n = \{x \in \{0, 1\}^n \mid \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq p + \frac{\varepsilon(n)}{2}\}$$

We have to show

$$|S_n| \geq \frac{\varepsilon(n)}{2} 2^n$$

Now,

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)]$$

$$= \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r) \cap x \in S_n] + \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r) \cap x \notin S_n]$$

$$= \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r) | x \in S_n] \Pr_x [x \in S_n] +$$

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r) | x \notin S_n] \Pr_x [x \notin S_n]$$

$$\leq \Pr_x [x \in S_n] + \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r) | x \notin S_n]$$

$$\Rightarrow \Pr_x [x \in S_n] \geq \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] - \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r) | x \notin S_n]$$

$$\text{Since, } \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq p + \varepsilon(n)$$

$$\text{and, } \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r) | x \notin S_n] < p + \frac{\varepsilon(n)}{2}$$

$$\text{so, } \Pr_x [x \in S_n] \geq (p + \varepsilon(n)) - \left(p + \frac{\varepsilon(n)}{2}\right)$$

$$\Rightarrow \Pr_x [x \in S_n] \geq \frac{\varepsilon(n)}{2}$$

$$\Rightarrow \text{Expected value of } |S_n| \geq \frac{\varepsilon(n)}{2} 2^n$$

Since, $x \xleftarrow{U} \{0, 1\}^n$ ■

Alternative proof

In this proof we shall use the results of section 6.2

Define a new random variable, Y as follows.

$$Y = \Pr[A(f(x), r) = B(x, r)]$$

$\implies 0 < Y < 1.$

Given , $\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq p + \varepsilon(n)$

Therefore, $Y \geq p + \varepsilon(n)$

Now,

$$\begin{aligned} E(Y) &= \sum_y y \cdot \Pr[Y = y] \\ &\geq (p + \varepsilon(n)) \sum_y \Pr[Y = y] \end{aligned}$$

$$\text{Since, } \sum_y \Pr[Y = y] = 1$$

$$\implies E(Y) \geq (p + \varepsilon(n))$$

$$\text{So, } \Pr \left[Y < (p + \varepsilon) - \frac{\varepsilon}{2} \right] = \Pr \left[1 - Y > 1 - (p + \varepsilon) + \frac{\varepsilon}{2} \right]$$

Observe the fact that since ε is negligible so $1 - (p + \varepsilon) + \frac{\varepsilon}{2} > 0$

now use markov inequality from 6.2.1 on $(1 - Y) > 0$, to get

$$\Pr \left[Y < (p + \varepsilon) - \frac{\varepsilon}{2} \right] \leq \frac{E(1 - Y)}{1 - (p + \varepsilon) + \frac{\varepsilon}{2}} = \frac{1 - E(Y)}{1 - (p + \varepsilon) + \frac{\varepsilon}{2}} \leq \frac{1 - (p + \varepsilon)}{1 - (p + \varepsilon) + \frac{\varepsilon}{2}} = 1 - \frac{\frac{\varepsilon}{2}}{1 - (p + \varepsilon) + \frac{\varepsilon}{2}}$$

$$\text{Since, } 0 < 1 - (p + \varepsilon) + \frac{\varepsilon}{2} < 1, \Pr \left[Y < (p + \varepsilon) - \frac{\varepsilon}{2} \right] \leq 1 - \frac{\frac{\varepsilon}{2}}{1 - (p + \varepsilon) + \frac{\varepsilon}{2}} < 1 - \frac{\varepsilon}{2}$$

$$\text{Hence, } \Pr \left[Y \geq (p + \varepsilon(n)) - \frac{\varepsilon(n)}{2} \right] \geq \frac{\varepsilon(n)}{2}$$

By definition , $S_n = \{x \in \{0, 1\}^n \mid \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq p + \frac{\varepsilon(n)}{2} \}$

$$\implies S_n = \{x \in \{0, 1\}^n \mid Y \geq p + \frac{\varepsilon(n)}{2} \}$$

$$\implies \Pr_x [x \in S_n] = \Pr \left[Y \geq p + \frac{\varepsilon(n)}{2} \right] \geq \frac{\varepsilon(n)}{2}$$

Since any value from $\{0, 1\}^n$ can belong to S_n with the probability $\frac{\varepsilon(n)}{2}$

And there are total 2^n values in $\{0, 1\}^n$

Hence ,expected value of $|S_n| \geq \frac{\varepsilon(n)}{2} 2^n$ ■

6.4.3 Theorem

$$\text{If } \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq \frac{3}{4} + \varepsilon(n)$$

Then \exists a set $S_n \subseteq \{0, 1\}^n$ of size at least $\frac{\varepsilon(n)}{2} 2^n$, where $\forall x \in \{0, 1\}^n$ such that

$$\Pr_{(r,x) \in \{0,1\}^n} [(A(F(x), r) = B(x, r)) \wedge (A(F(x), r \oplus e_i) = B(x, r \oplus e_i))] \geq \frac{1}{2} + \varepsilon(n)$$

Proof

We apply theorem 6.4.2 with the parameter $p = \frac{3}{4}$, then $\forall x \in S_n$

$$\begin{aligned} \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] &\geq \frac{3}{4} + \frac{\varepsilon(n)}{2} \\ \Rightarrow \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) \neq B(x, r)] &< \frac{1}{4} - \frac{\varepsilon(n)}{2} \end{aligned}$$

Also if i is fixed then, $r \xleftarrow{U} \{0, 1\}^n \Rightarrow r \oplus e_i \xleftarrow{U} \{0, 1\}^n$

$$\text{Then, } \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r \oplus e_i) \neq B(x, r \oplus e_i)] < \frac{1}{4} - \frac{\varepsilon(n)}{2},$$

Let us define the following events,

$$X_1 : A(F(X_1), r) \neq B(X_1, r)$$

$$X_2 : A(F(X_2), r \oplus e_i) \neq B(X_2, r \oplus e_i)$$

Then,

$$\Pr[X_1 = x] \leq \frac{1}{4} - \frac{\varepsilon(n)}{2}$$

$$\Pr[X_2 = x] \leq \frac{1}{4} - \frac{\varepsilon(n)}{2}$$

Now,

$$\begin{aligned} \Pr[X_1 = x \cup X_2 = x] &= \Pr[X_1 = x] + \Pr[X_2 = x] - \Pr[X_1 = x \cap X_2 = x] \\ &\leq \Pr[X_1 = x] + \Pr[X_2 = x] \\ &\leq \left(\frac{1}{4} - \frac{\varepsilon(n)}{2} \right) + \left(\frac{1}{4} - \frac{\varepsilon(n)}{2} \right) \\ &= \frac{1}{2} - \varepsilon(n) \end{aligned}$$

Hence,

$$\begin{aligned}
& \Pr_{(r,x) \in \{0,1\}^n} [(A(F(x), r) = B(x, r)) \wedge (A(F(x), r \oplus e_i) = B(x, r \oplus e_i))] \\
&= \Pr_{(r,x) \in \{0,1\}^n} [\overline{(X_1 = x)} \cap \overline{(X_2 = x)}] \\
&= 1 - \Pr_{(r,x) \in \{0,1\}^n} [X_1 = x \cup X_2 = x] \\
&\geq 1 - \left(\frac{1}{2} + \varepsilon(n)\right) = \frac{1}{2} + \varepsilon(n)
\end{aligned}$$

Hence proved ■

Now we present the final result of this section, namely the case 2 of the G-L theorem.

6.4.4 Statement

If F is an OWF then there is no such PPT algorithm A for which

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq \frac{3}{4} + \varepsilon(n)$$

Proof

We have a PPT algorithm A such that,

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] > \frac{3}{4} + \varepsilon(n)$$

We define another algorithm A' as follows -

$A'(F(x))$ {

For (i = 1 to n){

1. Choose a random $r \in \{0, 1\}^n$ uniformly and calculate

$$x_i = A(F(x), r) \oplus A(F(x), r \oplus e_i)$$

2. Repeat step 1 for k number of times and return the majority as the correct guess.

}
}

Define the following random variables,

$$M_j = 1, \text{ if } x_i = A(F(x), r) \oplus A(F(x), r \oplus e_i) \text{ is correct in } j \text{ th trial}$$

$$= 0, \text{ otherwise.}$$

$$j = 1, 2, \dots, k$$

So, $\Pr[M_j = 1] = p > \frac{1}{2} + \varepsilon(n)$, using theorem 6.4.3

now, $\Pr[A' \text{ gives wrong result in case of guessing a bit}]$

$$= \Pr[\text{maximum of } M_j \text{ s have value 0}]$$

$$= \Pr\left[\sum_j M_j \leq \frac{k}{2}\right]$$

Since M_j s are independent and $p \approx \frac{1}{2}$ then using chernoff's bound,

$$\Pr\left[\sum_j M_j \leq \frac{k}{2}\right] \leq \Pr\left[\left|\frac{\sum_{j=1}^k M_j}{k} - \left(\frac{1}{2} + \varepsilon\right)\right| > \varepsilon\right] < 2e^{-\frac{\varepsilon^2}{2\left(\frac{1}{2} + \varepsilon\right)\left(1 - \left(\frac{1}{2} - \varepsilon\right)\right)} \cdot k} = 2e^{-\frac{\varepsilon^2}{2\left(\frac{1}{4} - \varepsilon^2\right)} \cdot k} < 2e^{-2\varepsilon^2 k}$$

Now if we put,

$$k = \frac{\ln 4n}{2\varepsilon^2} = a \text{ polynomial in } n$$

$$\Pr\left[\sum_j M_j \leq \frac{k}{2}\right] < 2e^{-(\ln 4n)} = \frac{1}{2n}$$

Which implies A' is a PPT algorithm since it uses A polynomial times.

So, $\Pr[A' \text{ will guess wrong about one bit of } x \in S_n] < \frac{1}{2n}$

$\implies \Pr[A' \text{ will guess wrong about at least one bit of } x \in S_n] < n \cdot \frac{1}{2n} = \frac{1}{2}$

$\implies \Pr[A' \text{ will guess correctly about all bits of } x \in S_n] \geq \frac{1}{2}$

$\implies \Pr[A' \text{ will guess correctly about all bits of } x] \geq$

$$\Pr[A' \text{ will guess correctly about all bits of } x \in S_n] \times \Pr[x \in S_n]$$

$$> \frac{1}{2} \times \frac{\varepsilon}{2}, \text{ from theorem 6.4.2}$$

$$\Rightarrow \Pr_{(r,x) \in \{0,1\}^n} [A'(F(x), r) = x] > \frac{\varepsilon(n)}{4}$$

Which is a contradiction to the definition of OWF since A' is a PPT algorithm

Hence our primary assumption was wrong and no such PPT algorithm A' exists

Hence proved. ■

6.5 Proof of G-L Theorem: Case 3

Now we lower the probability bound from 1 to $(\frac{1}{2} + \varepsilon(n))$ on the PPT algorithm A which can now compute $B(x, r)$ from $F(x)$ and r with a probability more than $\frac{1}{2} + \varepsilon(n)$.

In that case we have to prove the existence of another PPT algorithm A' which can compute x from $F(x)$ only with a high probability.

So we are given a PPT algorithm A such that

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] > \frac{1}{2} + \varepsilon(n)$$

We shall use a little different strategy here to construct A' from A compare to 6.4. In 6.4 we computed two $B(\cdot)$ values in each trial which effectively doubles the error probability $\varepsilon(n)$.

Thus this A' does not actually invert x given $F(x)$ with a significant probability. Hence in this proof we use just guess just one $B(\cdot)$ value using A in each trial and we shall compute the other $B(\cdot)$ value as a combination of some pre-computed $B(\cdot)$ values.

Next we define some notations which are instrumental in proving the result of this section. Then we shall carry on with the proof.

1. Set $m = h(n)$ and $l = \log_2(m + 1)$, later we prove $h = \text{polynomial}(n)$
2. Choose l strings uniformly and randomly from $\{0, 1\}^n$ and denote them as s^1, s^2, \dots, s^l
3. Guess $B(x, s^i)$, $i = 1, 2, \dots, l$ using the PPT algorithm A and denote those guesses as $\sigma^1, \sigma^2, \dots, \sigma^l$
4. Since, $\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] > \frac{1}{2} + \varepsilon(n)$

So, \Pr [all $\sigma^1, \sigma^2, \dots, \sigma^l$ are correct guesses]

$$= \Pr_{(r,x) \in \{0,1\}^n} \left[\bigcap_{j=1}^l (A(F(x), s^j) = B(x, s^j)) \right]$$

$$\begin{aligned}
&= \prod_{j=1}^l \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), s^j) = B(x, s^j)] \\
&> \left(\frac{1}{2} + \varepsilon(n)\right)^l \approx \frac{1}{2^l} = \frac{1}{m+1} = \frac{1}{\text{polynomial}(n)}
\end{aligned}$$

5. Let $J \subseteq \{0, 1, \dots, l\}$, define, $r^J = \bigoplus_{j \in J} s^j$

6. Observe $B(x, r^J) = B\left(x, \bigoplus_{j \in J} s^j\right) = \bigoplus_{j \in J} B(x, s^j) = \bigoplus_{j \in J} \sigma^j$ hence we can say that given σ^j 's we can compute $\rho^J = \bigoplus_{j \in J} \sigma^j$ as our guess for $B(x, r^J)$

7. **Claim:** r^J 's are pair wise independent and uniformly distributed over $\{0, 1\}^n$

Proof

For all unequal subsets J and K there exist $j \in J$ and $k \in K - J$

Hence for every $\alpha, \beta \in \{0, 1\}^n$

$$\begin{aligned}
\Pr[r^K = \beta | r^J = \alpha] &= \Pr[s^k = \beta | s^j = \alpha] \\
&= \Pr[s^k = \beta] \\
&= \Pr[r^K = \beta]
\end{aligned}$$

Hence r^J 's are pair wise independent.

Observe that if $r^J = \alpha$ then we can have any choice of strings for the first $(|J| - 1)$ elements of r^J and the last string will be fixed accordingly.

$$\text{So, } \Pr[r^J = \alpha] = \frac{(2^n)^{|J|-1}}{(2^n)^{|J|}} = \frac{1}{2^n}$$

Hence r^J 's are uniformly distributed over $\{0, 1\}^n$ ■

6.5.1 Theorem

$$\text{If } \Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq \frac{1}{2} + \varepsilon(n)$$

Then \exists a set $S_n \subseteq \{0, 1\}^n$ of size at least $\frac{\varepsilon(n)}{2} 2^n$, where $\forall x \in \{0, 1\}^n$

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq \frac{1}{2} + \frac{\varepsilon(n)}{2}$$

Proof

The proof is exactly similar as 6.4.2 with $p = \frac{1}{2}$ ■

6.5.2 Theorem

$\forall x \in S_n$ and $1 \leq i \leq n$

$$\Pr \left[\left| \left\{ J \mid B(x, r^J) \oplus A(F(x), r^J \oplus e_i) = x_i \right\} \right| > \frac{1}{2} (2^l - 1) \right] > 1 - \frac{1}{2n}$$

Proof

$\forall J \subseteq \{1, 2, \dots, l\}$ define a random variable $M^J \in \{0, 1\}$

$$M^J = 1 \text{ iff } B(x, r^J) \oplus A(F(x), r^J \oplus e_i) = B(x, e_i) = x_i$$

= 0 otherwise

$$\Rightarrow M^J = 1 \text{ iff } A(F(x), r^J \oplus e_i) = B(x, r^J \oplus e_i)$$

$$\Rightarrow \Pr [M^J = 1] \geq \frac{1}{2} + \frac{\varepsilon(n)}{2}, \text{ where } x \in S_n$$

now, there may be $(2^l - 1)$ different J s and $(2^l - 1) = m$

$$\Rightarrow \Pr \left[\left| \left\{ J \mid B(x, r^J) \oplus A(F(x), r^J \oplus e_i) = x_i \right\} \right| > \frac{1}{2} (2^l - 1) \right]$$

$$= \Pr \left[\sum_J M^J > \frac{1}{2} (2^l - 1) \right] = \Pr \left[\sum_J M^J > \frac{m}{2} \right]$$

$$\text{now, } E(M^J) = 1 \cdot \Pr[M^J = 1] + 0 \cdot \Pr[M^J = 0]$$

$$\geq \left(\frac{1}{2} + \frac{\varepsilon(n)}{2} \right)$$

$$\text{thus } E \left(\sum_J M^J \right) = \sum_J E(M^J) \geq \left(\frac{1}{2} + \frac{\varepsilon(n)}{2} \right) m$$

$$\text{Var}(M^J) = E \left((M^J)^2 \right) - (E(M^J))^2$$

$$= 1^2 \cdot \Pr[M^J = 1] + 0^2 \cdot \Pr[M^J = 0] - (\Pr[M^J = 1])^2$$

$$= (\Pr[M^J = 1]) (1 - \Pr[M^J = 1])$$

Since the function $f(x) = x - x^2$ is decreasing in the domain $\frac{1}{2} \leq x < 1$

$$\text{thus for } M^J \geq \left(\frac{1}{2} + \frac{\varepsilon(n)}{2} \right)$$

$$\text{Var}(M^J) \leq \left(\frac{1}{2} + \frac{\varepsilon(n)}{2} \right) \left(1 - \left(\frac{1}{2} + \frac{\varepsilon(n)}{2} \right) \right)$$

Now M^J depends on the choice of r^J . Since r^J 's are chosen pair wise independently so M^J s are also pairwise independent.

$$\begin{aligned} \text{Hence, } \text{Var} \left(\sum_J M^J \right) &= \sum_J \text{Var} (M^J) \quad [\text{Since, } \text{Cov} (M^I, M^K) = 0 \forall I, K] \\ &\leq m \left(\frac{1}{2} + \frac{\varepsilon(n)}{2} \right) \left(\frac{1}{2} - \frac{\varepsilon(n)}{2} \right) < \frac{m}{4} \end{aligned}$$

$$\begin{aligned} \text{now, } \Pr \left[\sum_J M^J \leq \frac{m}{2} \right] &\leq \Pr \left[\left| \sum_J M^J - \left(\frac{1}{2} + \frac{\varepsilon(n)}{2} \right) m \right| \leq \frac{\varepsilon(n)}{2} m \right] \\ &\leq \frac{\text{Var} \left(\sum_J M^J \right)}{\left(\frac{\varepsilon(n)}{2} m \right)^2} \quad (\text{by Chebyshev's Inequality from 6.2.2}) \\ &< \frac{\frac{m}{4}}{\left(\frac{\varepsilon(n)}{2} m \right)^2} = \frac{1}{(\varepsilon(n))^2 m} \end{aligned}$$

if we take $m = \frac{2n}{(\varepsilon(n))^2}$

$$\begin{aligned} \Pr \left[\sum_J M^J \leq \frac{m}{2} \right] &\leq \frac{1}{2n} \\ \Rightarrow \Pr \left[\sum_J M^J > \frac{m}{2} \right] &\leq 1 - \frac{1}{2n} \end{aligned}$$

Hence proved. ■

Next we present the final result of this section, and also of this chapter, namely the case 3 of the G-L theorem that is proof for the most general case.

6.5.3 Statement

If F is an OWF then there is no such PPT algorithm A for which

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] \geq \frac{1}{2} + \varepsilon(n)$$

Proof

Suppose we have a PPT algorithm A such that,

$$\Pr_{(r,x) \in \{0,1\}^n} [A(F(x), r) = B(x, r)] > \frac{1}{2} + \varepsilon(n)$$

We define another algorithm A' as follows -

$A'(F(x)) \{$

1. Generate $s^1, s^2, \dots, s^l \in \{0, 1\}^n$ uniformly and independently and compute $\sigma^1, \sigma^2, \dots, \sigma^l$.
2. $\forall J \subseteq \{1, 2, \dots, l\} - \{\varphi\}$ compute $r^J = \bigoplus_{j \in J} s^j$ and $\rho^J = \bigoplus_{j \in J} \sigma^j$
3. $\forall J \subseteq \{1, 2, \dots, l\} - \{\varphi\}$ and $i = 1, 2, \dots, n$ compute

$$z_i^J = \rho^J \oplus A(F(x), r^J \oplus e_i)$$

4. For all $i = 1, 2, \dots, n$ set $z_i =$ majority of z_i^J values.
5. output $z = z_1 z_2 \dots z_n$

$\}$

Using theorem 6.5.2 z_i^J has a majority equal to x_i

$$\text{and also } \Pr[A' \text{ is wrong for } i \text{ th bit of } x] = \Pr\left[\sum_J M^J \leq \frac{m}{2}\right] \leq \frac{1}{2^n}$$

We define the following events,

$X_k : A'$ is wrong for the k th bit of x , $k = 1, 2, \dots, n$

Then $\Pr[A' \text{ is wrong for at least one } i]$

$$= \Pr[X_1 \cup X_2 \cup \dots \cup X_n] < \sum_{i=1}^n \Pr[X_i] < n \cdot \frac{1}{2^n} = \frac{1}{2}$$

$$\Rightarrow \Pr[A' \text{ is correct for all } i \text{ values}] > \frac{1}{2}$$

Again from section 6.5

$$\Pr[\text{all } \sigma^1, \sigma^2, \dots, \sigma^l \text{ are correct guesses}] \approx \frac{1}{2^l}$$

$$\text{From theorem 6.5.1, } \Pr[x \in S_n] \geq \frac{\varepsilon(n)}{2}$$

$$\text{Again from theorem 6.5.2, } m = \frac{2n}{(\varepsilon(n))^2}$$

$$\Pr[A'(F(x)) = x \cap x \in S_n]$$

$$= \Pr[\text{all } \sigma^1, \sigma^2, \dots, \sigma^l \text{ are correct guesses} \cap$$

$$A' \text{ is correct for all } i \text{ values} \cap x \in S_n]$$

$$> \frac{1}{2^l} \times \frac{1}{2} \times \frac{\varepsilon(n)}{2}$$

$$\begin{aligned}
&= \frac{\varepsilon(n)}{4} \times \frac{1}{(m+1)} \\
&= \frac{\varepsilon(n)}{4} \times \frac{1}{\left(\frac{2n}{\varepsilon(n)^2} + 1\right)}
\end{aligned}$$

Now, $\Pr[A'(F(x)) = x]$

$$= \Pr[A'(F(x)) = x \cap x \in S_n] + \Pr[A'(F(x)) = x \cap x \notin S_n]$$

$$\geq \Pr[A'(F(x)) = x \cap x \in S_n]$$

$$> \frac{\varepsilon(n)}{4} \times \frac{1}{\left(\frac{2n}{\varepsilon(n)^2} + 1\right)}$$

Since $\varepsilon(n) = \frac{1}{p(n)}$, $p(n) =$ Some polynomial in n

Ultimately we have the result

$$\Pr[A'(F(x)) = x] > \frac{1}{4} \times \frac{1}{\left(p(n) + 2n(p(n))^3\right)}$$

Now A' is a PPT algorithm, since it invokes the PPT algorithm A $\frac{2n}{\varepsilon(n)^2}$ or $2n(p(n))^2$ time.

Hence by definition f cannot be a strong OWF.

Which is a contradiction.

So no such PPT algorithm A exists.

We can conclude that $B(x, r)$ is a hardcore predicate. ■