SANJAM GARG

A COURSE IN THEORY OF CRYPTOGRAPHY

This draft was compiled on Tuesday 3rd September, 2024.

Contents

Preface

Cryptography enables many paradoxical objects, such as public key encryption, verifiable electronic signatures, zero-knowledge protocols, and fully homomorphic encryption. The two main steps in developing such seemingly impossible primitives are (i) defining the desired security properties formally and (ii) obtaining a construction satisfying the security property provably. In modern cryptography, the second step typically assumes (unproven) computational assumptions, which are conjectured to be computationally intractable. In this course, we will define several cryptographic primitives and argue their security based on well-studied computational hardness assumptions. However, we will largely ignore the mathematics underlying the assumed computational intractability assumptions.

Acknowledgements

These lecture notes are based on scribe notes taken by students in CS 276 over the years. Also, thanks to Peihan Miao, Akshayaram Srinivasan, and Bhaskar Roberts for helping to improve these notes.

1
Mathematical Background

In modern cryptography, (1) we typically assume that our attackers cannot run in unreasonably large amounts of time, and (2) we allow security to be broken with a *very small*, but non-zero, probability.

Without these assumptions, we must work in the realm of information-theoretic cryptography, which is often unachievable or impractical for many applications. For example, the one-time pad [1] – an information-theoretically secure cipher – is not very useful because it requires very large keys.

[1] For a message $m \in \{0,1\}^n$ and a random key $k \in \{0,1\}^n$, the encryption of m is $c = m \oplus k$. The decryption is $m = c \oplus k$.

In this chapter, we define items (1) and (2) more formally. We require our adversaries to run in polynomial time, which captures the idea that their runtime is not unreasonably large (sections 1.1). We also allow security to be broken with negligible – very small – probability (section 1.2).

1.1 Probabilistic Polynomial Time

A probabilistic Turing Machine is a generic computer that is allowed to make random choices during its execution. A probabilistic *polynomial time* Turing Machine is one which halts in time polynomial in its input length. More formally:

Definition 1.1 (Probabilistic Polynomial Time). *A probabilistic Turing Machine M is said to be PPT (a Probabilistic Polynomial Time Turing Machine) if $\exists c \in \mathbb{Z}^+$ such that $\forall x \in \{0,1\}^*$, $M(x)$ halts in $|x|^c$ steps.*

A *non-uniform* PPT Turing Machine is a collection of machines one for each input length, as opposed to a single machine that must work for all input lengths.

Definition 1.2 (Non-uniform PPT). *A non-uniform PPT machine is a sequence of Turing Machines $\{M_1, M_2, \cdots\}$ such that $\exists c \in \mathbb{Z}^+$ such that $\forall x \in \{0,1\}^*$, $M_{|x|}(x)$ halts in $|x|^c$ steps.*

1.2 Noticeable and Negligible Functions

Noticeable and negligible functions are used to characterize the "largeness" or "smallness" of a function describing the probability of some event. Intuitively, a noticeable function is required to be larger than some inverse-polynomially function in the input parameter. On the other hand, a negligible function must be smaller than any inverse-polynomial function of the input parameter. More formally:

Definition 1.3 (Noticeable Function)**.** *A function $\mu(\cdot) : \mathbb{Z}^+ \to [0,1]$ is noticeable iff $\exists c \in \mathbb{Z}^+, n_0 \in \mathbb{Z}^+$ such that $\forall n \geq n_0,\ \mu(n) > n^{-c}$.*

Example. Observe that $\mu(n) = n^{-3}$ is a noticeable function. (Notice that the above definition is satisfied for $c = 4$ and $n_0 = 1$.)

Definition 1.4 (Negligible Function)**.** *A function $\mu(\cdot) : \mathbb{Z}^+ \to [0,1]$ is negligible iff $\forall c \in \mathbb{Z}^+\ \exists n_0 \in \mathbb{Z}^+$ such that $\forall n \geq n_0,\ \mu(n) < n^{-c}$.*

Example. $\mu(n) = 2^{-n}$ is an example of a negligible function. This can be observed as follows. Consider an arbitrary $c \in \mathbb{Z}^+$ and set $n_0 = c^2$. Now, observe that for all $n \geq n_0$, we have that $\frac{n}{\log_2 n} \geq \frac{n_0}{\log_2 n_0} > \frac{n_0}{\sqrt{n_0}} = \sqrt{n_0} = c$. This allows us to conclude that

$$\mu(n) = 2^{-n} = n^{-\frac{n}{\log_2 n}} < n^{-c}.$$

Thus, we have proved that for any $c \in \mathbb{Z}^+$, there exists $n_0 \in \mathbb{Z}^+$ such that for any $n \geq n_0$, $\mu(n) < n^{-c}$.

Gap between Noticeable and Negligible Functions. At first thought it might seem that a function that is not negligible (or, a non-negligible function) must be a noticeable. This is not true![2] Negating the definition of a negligible function, we obtain that a non-negligible function $\mu(\cdot)$ is such that $\exists c \in \mathbb{Z}^+$ such that $\forall n_0 \in \mathbb{Z}^+$, $\exists n \geq n_0$ such that $\mu(n) > n^{-c}$. Note that this requirement is satisfied as long as $\mu(n) > n^{-c}$ for infinitely many choices of $n \in \mathbb{Z}^+$. However, a noticeable function requires this condition to be true for every $n \geq n_0$.

[2] Mihir Bellare. A note on negligible functions. *Journal of Cryptology*, 15 (4):271–284, September 2002. DOI: 10.1007/s00145-002-0116-x

Below we give example of a function $\mu(\cdot)$ that is neither negligible nor noticeable.

$$\mu(n) = \begin{cases} 2^{-n} & : x \mod 2 = 0 \\ n^{-3} & : x \mod 2 \neq 0 \end{cases}$$

This function is obtained by interleaving negligible and noticeable functions. It cannot be negligible (resp., noticeable) because it is greater (resp., less) than an inverse-polynomially function for infinitely many input choices.

Properties of Negligible Functions. Sum and product of two negligible functions is still a negligible function. We argue this for the sum function below and defer the problem for products to Exercise 2.2. These properties together imply that any polynomial function of a negligible function is still negligible.

Exercise 1.1. *If $\mu(n)$ and $\nu(n)$ are negligible functions from domain $\mathbb{Z}^+$ to range $[0,1]$ then prove that the following functions are also negligible:*

1. $\psi_1(n) = \frac{1}{2} \cdot (\mu(n) + \nu(n))$
2. $\psi_2(n) = \min\{\mu(n) + \nu(n), 1\}$
3. $\psi_3(n) = \mu(n) \cdot \nu(n)$
4. $\psi_4(n) = \mathsf{poly}(\mu(n))$, *where* $\mathsf{poly}(\cdot)$ *is an unspecified polynomial function. (Assume that the output is also clamped to* $[0,1]$ *to satisfy the definition)*

function.

Proof.

1. We need to show that for any $c \in \mathbb{Z}^+$, we can find n_0 such that $\forall n \geq n_0$, $\psi_1(n) \leq n^{-c}$. Our argument proceeds as follows. Given the fact that μ and ν are negligible we can conclude that there exist n_1 and n_2 such that $\forall n \geq n_1$, $\mu(n) < n^{-c}$ and $\forall n \geq n_2$, $g(n) < n^{-c}$. Combining the above two facts and setting $n_0 = \max(n_1, n_2)$ we have that for every $n \geq n_0$,

$$\psi_1(n) = \frac{1}{2} \cdot (\mu(n) + \nu(n)) < \frac{1}{2} \cdot (n^{-c} + n^{-c}) = n^{-c}$$

   Thus, $\psi_1(n) \leq n^{-c}$ and hence is negligible.

2. We need to show that for any $c \in \mathbb{Z}^+$, we can find n_0 such that $\forall n \geq n_0$, $\psi_2(n) \leq n^{-c}$. Given the fact that μ and ν are negligible, there exist n_1 and n_2 such that $\forall n \geq n_1$, $\mu(n) \leq n^{-c-1}$ and $\forall n \geq n_2$, $g(n) \leq n^{-c-1}$. Setting $n_0 = \max(n_1, n_2, 3)$ we have that for every $n \geq n_0$,

$$\psi_2(n) = \min\{\mu(n) + \nu(n), 1\} < n^{-c-1} + n^{-c-1} < n^{-c}$$

□

2 One-Way Functions

Cryptographers often attempt to base cryptographic results on conjectured computational assumptions to leverage reduced adversarial capabilities. Furthermore, the security of these constructions is no better than the assumptions they are based on.

> *Cryptographers seldom sleep well.*[1]

[1] Quote by Silvio Micali in personal communication with Joe Kilian.

Thus, basing cryptographic tasks on the *minimal* necessary assumptions is a key tenet in cryptography. Towards this goal, rather can making assumptions about specific computational problem in number theory, cryptographers often consider *abstract primitives*. The existence of these abstract primitives can then be based on one or more computational problems in number theory.

The weakest abstract primitive cryptographers consider is one-way functions. Virtually, every cryptographic goal of interest is known to imply the existence of one-way functions. In other words, most cryptographic tasks would be impossible if the existence of one-way functions was ruled out. On the flip side, the realizing cryptographic tasks from just one-way functions would be ideal.

2.1 Definition

A one-way function $f : \{0,1\}^n \to \{0,1\}^m$ is a function that is easy to compute but hard to invert. This intuitive notion is trickier to formalize than it might appear on first thought.

Definition 2.1 (One-Way Functions)**.** *A function $f : \{0,1\}^* \to \{0,1\}^*$ is said to be one-way function if:*

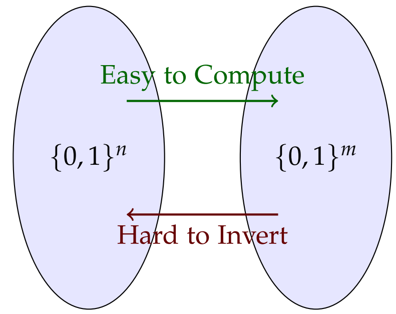
- ***Easy to Compute:*** *$\exists$ a (deterministic) polynomial time machine M such that $\forall x \in \{0,1\}^*$ we have that*

$$M(x) = f(x)$$

- ***Hard to Invert:*** *$\forall$ non-uniform PPT adversary $\mathcal{A}$ we have that*

$$\mu_{\mathcal{A},f}(n) = \Pr_{x \xleftarrow{\$} \{0,1\}^n} [\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))] \tag{2.1}$$

is a negligible function, $x \xleftarrow{\$} \{0,1\}^n$ denotes that x is drawn uniformly at random from the set $\{0,1\}^n$, $f^{-1}(f(x)) = \{x' \mid f(x) = f(x')\}$, and the probability is over the random choices of x and the random coins of $\mathcal{A}$[2].



We note that the function is not necessarily one-to-one. In other words, it is possible that $f(x) = f(x')$ for $x \neq x'$ – and the adversary is allowed to output any such x'.

The above definition is rather delicate. We next describe problems in the slight variants of this definition that are insecure.

1. What if we require that $\Pr_{x \xleftarrow{\$} \{0,1\}^n} [\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))] = 0$ instead of being negligible?

   This condition is false for every function f. An adversary $\mathcal{A}$ that outputs an arbitrarily fixed value x_0 succeeds with probability at least $1/2^n$, as $x_0 = x$ with at least the same probability.

2. What if we drop the input 1^n to $\mathcal{A}$ in Equation 2.1?

   Consider the function $f(x) = |x|$. In this case, we have that $m = \log_2 n$, or $n = 2^m$. Intuitively, f should not be considered a one-way function, because it is easy to invert f. Namely, given a value y any x such that $|x| = y$ is such that $x \in f^{-1}(y)$. However, according to this definition the adversary gets an m bit string as input, and hence is restricted to running in time polynomial in m. Since each possible x is of size $n = 2^m$, the adversary doesn't even have enough time to write down the answer! Thus, according to the flawed definition above, f would be a one-way function.

   Providing the attacker with 1^n (n repetitions of the 1 bit) as additional input avoids this issue. In particular, it allows the attacker to run in time polynomial in m and n.

Candidate One-way Functions. It is not known whether one-way functions exist. In fact, the existence of one-way functions would

[2] Typically, the probability is only taken over the random choices of x, since we can fix the random coins of the adversary $\mathcal{A}$ that maximize its advantage.

imply that $P \neq NP$ (see Exercise 2.3).

However, there are candidates of functions that could be one-way functions, based on the difficulty of certain computational problems. One example is based on the hardness of factoring. Multiplication can be done easily in $O(n^2)$ time, but so far no polynomial time algorithm is known for factoring. Explicitly, we can define the function $f_1 : P_n \times P_n \to \mathbb{Z}$ where P_n is the set of all n-bit primes as $f_1(p, q) = p \cdot q$.

Another candidate is based on the hardness of the discrete logarithm problem. Given a group $\mathbb{G}$ of prime order q and a generator g, the discrete logarithm problem is to find x such that $g^x = y$ for a given y. The function $f_2 : \mathbb{Z}_q \to \mathbb{G}$ defined as $f_2(x) = g^x$ is also believed to be one-way assuming the hardness of the discrete logarithm problem.

2.2 *Robustness and Brittleness of One-way Functions*

What operations can we perform on one-way functions and still have a one-way function? In this section, we explore the robustness and brittleness of one-way functions and some operations that are safe or unsafe to perform on them.

2.2.1 *Robustness*

Consider having a one-way function f. Can we use this function f in order to make a more structured one-way function g such that $g(x_0) = y_0$ for some constants x_0, y_0, or would this make the function no longer be one-way?

Intuitively, the answer is yes - we can specifically set $g(x_0) = y_0$, and otherwise have $g(x) = f(x)$. In this case, the adversary gains the knowledge of how to invert y_0, but that will only happen with negligible probability, and so the function is still one-way.

In fact, this can be done for an exponential number of x_0, y_0 pairs. To illustrate that, consider the following function:

$$g(x_1 \| x_2) = \begin{cases} x_1 \| x_2 & : x_1 = 0^{n/2} \\ f(x_1 \| x_2) & : \text{otherwise} \end{cases}$$

However, this raises an apparent contradiction - according to this theorem, given a one-way function f, we could keep fixing each of its values to 0, and it would continue to be a one-way function. If we kept doing this, we would eventually end up with a function which outputs 0 for *all* of the possible values of x. How could this still be one-way?

The resolution of this apparent paradox is by noticing that a one-way function is only required to be one-way in the limit where n grows very large. So, no matter how many times we fix the values of f to be 0, we are still only setting a finite number of x values to 0. However, this will still satisfy the definition of a one-way function - it is just that we will have to use larger and larger values of n_0 in order to prove that the probability of breaking the one-way function is negligible.

2.2.2 *Brittleness*

Example: OWFs do not always compose securely. Given a one-way function $f : \{0,1\}^n \to \{0,1\}^n$, is the function $f^2(x) = f(f(x))$ also a one-way function? Intuitively, it seems that if it is hard to invert $f(x)$, then it would be just as hard to invert $f(f(x))$. However, this intuition is incorrect and highlights the delicacy when working with cryptographic assumptions and primitives. In particular, assuming one-way functions exists we describe a one-way function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ such that f^2 can be efficiently inverted. Let $g : \{0,1\}^n \to \{0,1\}^n$ be a one-way function then we set f as follows:

$$f(x_1, x_2) = 0^n \| g(x_1)$$

Two observations follow:

1. f^2 is not one-way. This follows from the fact that for all inputs x_1, x_2 we have that $f^2(x_1, x_2) = 0^{2n}$. This function is clearly not one-way!

2. f is one-way. This can be argued as follows. Assume that there exists an adversary $\mathcal{A}$ such that $\mu_{\mathcal{A},f}(n) = \Pr_{x \xleftarrow{\$} \{0,1\}^n}[\mathcal{A}(1^{2n}, f(x)) \in f^{-1}(f(x))]$ is non-negligible. Using such an $\mathcal{A}$ we will describe a construction of adversary $\mathcal{B}$ such that $\mu_{\mathcal{B},g}(n) = \Pr_{x \xleftarrow{\$} \{0,1\}^n}[\mathcal{B}(1^n, g(x)) \in g^{-1}(g(x))]$ is also non-negligible. This would be a contradiction thus proving our claim.

   Description of $\mathcal{B}$: $\mathcal{B}$ on input $y \in \{0,1\}^n$ outputs the n lower-order bits of $\mathcal{A}(1^{2n}, 0^n \| y)$.

   Observe that if $\mathcal{A}$ successfully inverts f then we have that $\mathcal{B}$ successfully inverts g. More formally, we have that:

$$\mu_{\mathcal{B},g}(n) = \Pr_{x \xleftarrow{\$} \{0,1\}^n} \left[\mathcal{A}(1^{2n}, 0^n || g(x)) \in \{0,1\}^n || g^{-1}(g(x)) \right].$$

But

$$\begin{aligned}\mu_{\mathcal{A},f}(2n) = &\Pr_{x_1,x_2 \xleftarrow{\$} \{0,1\}^{2n}}[\mathcal{A}(1^{2n}, f(x_1,x_2)) \in f^{-1}(f(\tilde{x}))] \\ = &\Pr_{x_1 \xleftarrow{\$} \{0,1\}^{n}}[\mathcal{A}(1^{2n}, 0^n||g(x_2)) \in \{0,1\}^n||g^{-1}(g(x_2))] \\ = &\mu_{\mathcal{B},g}(n).\end{aligned}$$

Hence, we have that $\mu_{\mathcal{B},g}(n) = \mu_{\mathcal{A},f}(2n)$ which is non-negligible as long as $\mu_{\mathcal{A},f}(2n)$ is non-negligible.

Example: Dropping a bit is not always secure. Below is another example of a transformation that does not work. Given any one-way function g, let $g'(x)$ be $g(x)$ with the first bit omitted.

Claim 2.1. *g' is not necessarily one-way. In other words, there exists a OWF function g for which g' is not one-way.*

Proof. We must (1) construct a function g, (2) show that g is one-way, and (3) show that g' is not one-way.

Step 1: Construct a OWF g. To do this, we first want to come up with a (contrived) function g and prove that it is one-way. Let us assume that there exists a one-way function $h : \{0,1\}^n \to \{0,1\}^n$. We define the function $g : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as follows:

$$g(x\|y) = \begin{cases} 0^n\|y & \text{if } x = 0^n \\ 1\|0^{n-1}\|g(y) & \text{otherwise} \end{cases}$$

Step 2: Prove that g is one-way.

Claim 2.2. *If h is a one-way function, then so is g.*

Proof. Assume for the sake of contradiction that g is not one-way. Then there exists a polynomial time adversary $\mathcal{A}$ and a non-negligible function $\mu(\cdot)$ such that:

$$\Pr_{x,y}[\mathcal{A}(1^n, g(x\|y)) \in g^{-1}(g(x\|y))] = \mu(n)$$

We will use such an adversary $\mathcal{A}$ to invert h with some non-negligible probability. This contradicts the one-wayness of h and thus our assumption that g is not one-way function is false.

Let us now construct an $\mathcal{B}$ that uses $\mathcal{A}$ and inverts h. $\mathcal{B}$ is given $1^n, h(y)$ for a randomly chosen y and its goal is to output $y' \in h^{-1}(h(y))$ with some non-negligible probability. $\mathcal{B}$ works as follows:

1. It samples $x \leftarrow \{0,1\}^n$ randomly.

2. If $x = 0^n$, it samples a random $y' \leftarrow \{0,1\}^n$ and outputs it.

3. Otherwise, it runs $\mathcal{A}(10^{n-1}\|h(y))$ and obtains $x'\|y'$. It outputs y'.

Let us first analyze the running time of $\mathcal{B}$. The first two steps are clearly polynomial (in n) time. In the third step, $\mathcal{B}$ runs $\mathcal{A}$ and uses its output. Note that the running time of since $\mathcal{A}$ runs in polynomial (in n) time, this step also takes polynomial (in n) time. Thus, the overall running time of $\mathcal{B}$ is polynomial (in n).

Let us now calculate the probability that $\mathcal{B}$ outputs the correct inverse. If $x = 0^n$, the probability that y' is the correct inverse is at least $\frac{1}{2^n}$ (because it guesses y' randomly and probability that a random y' is the correct inverse is $\geq 1/2^n$). On the other hand, if $x \neq 0^n$, then the probability that $\mathcal{B}$ outputs the correct inverse is $\mu(n)$. Thus,

$$\begin{aligned}
\Pr[\mathcal{B}(1^n, h(y)) \in h^{-1}(h(y))] &\geq \Pr[x = 0^n](\frac{1}{2^n}) + \Pr[x \neq 0^n]\mu(n) \\
&= \frac{1}{2^{2n}} + (1 - \frac{1}{2^n})\mu(n) \\
&\geq \mu(n) - (\frac{1}{2^n} - \frac{1}{2^{2n}})
\end{aligned}$$

Since $\mu(n)$ is a non-negligible function and $(\frac{1}{2^n} - \frac{1}{2^{2n}})$ is a negligible function, their difference is non-negligible.[3] This contradicts the one-wayness of h.

[3] Exercise: Prove that if $\alpha(\cdot)$ is a non-negligible function and $\beta(\cdot)$ is a negligible function, then $(\alpha - \beta)(\cdot)$ is a non-negligible function.

□

Step 3: Prove that g' is not one-way. We construct the new function $g' : \{0,1\}^{2n} \to \{0,1\}^{2n-1}$ by dropping the first bit of g. That is,

$$g'(x\|y) = \begin{cases} 0^{n-1}\|y & \text{if } x = 0^n \\ 0^{n-1}\|g(y) & \text{otherwise} \end{cases}$$

We now want to prove that g' is not one-way. That is, we want to design an adversary $\mathcal{C}$ such that given 1^{2n} and $g'(x\|y)$ for a randomly chosen x, y, it outputs an element in the set $g^{-1}(g(x\|y)$. The description of $\mathcal{C}$ is as follows:

- On input 1^{2n} and $g'(x\|y)$, the adversary $\mathcal{C}$ parses $g'(x\|y)$ as $0^{n-1}\|\overline{y}$.
- It outputs $0^n\|\overline{y}$ as the inverse.

Notice that $g'(0^n\|\overline{y}) = 0^{n-1}\|\overline{y}$. Thus, $\mathcal{C}$ succeeds with probability 1 and this breaks the one-wayness of g'.

□

2.3 Hardness Amplification

In this section, we show that even a very *weak* form of one-way functions suffices from constructing one-way functions as defined previously. For this section, we refer to this previously defined notion as strong one-way functions.

Definition 2.2 (Weak One-Way Functions)**.** *A function $f : \{0,1\}^n \to \{0,1\}^m$ is said to be a weak one-way function if:*

- *f is computable by a polynomial time machine, and*
- *There exists a noticeable function $\alpha_f(\cdot)$ such that $\forall$ non-uniform PPT adversaries $\mathcal{A}$ we have that*

$$\mu_{\mathcal{A},f}(n) = \Pr_{x \xleftarrow{\$} \{0,1\}^n} [\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))] \le 1 - \alpha_f(n).$$

Theorem 2.1. *If there exists a weak one-way function, then there exists a (strong) one-way function.*

Proof. We prove the above theorem constructively. Suppose $f : \{0,1\}^n \to \{0,1\}^m$ is a weak one-way function, then we prove that the function $g : \{0,1\}^{nq} \to \{0,1\}^{mq}$ for $q = \lceil \frac{2n}{\alpha_f(n)} \rceil$ where

$$g(x_1, x_2, \cdots, x_q) = f(x_1)||f(x_2)|| \cdots ||f(x_q),$$

is a strong one-way function.

Assume for the sake of contradiction that there exists an adversary $\mathcal{B}$ such that $\mu_{\mathcal{B},g}(nq) = \Pr_{x \xleftarrow{\$} \{0,1\}^{nq}} [\mathcal{B}(1^{nq}, g(x)) \in g^{-1}(g(x))]$ is non-negligible. Then we use $\mathcal{B}$ to construct $\mathcal{A}$ (see Figure 2.1) that breaks f, namely $\mu_{\mathcal{A},f}(n) = \Pr_{x \xleftarrow{\$} \{0,1\}^n} [\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))] > 1 - \alpha_f(n)$ for sufficiently large n.

1. $i \xleftarrow{\$} [q]$.
2. $x_1, \cdots, x_{i-1}, x_i, \cdots, x_q \xleftarrow{\$} \{0,1\}^n$.
3. Set $y_j = f(x_j)$ for each $j \in [q] \backslash \{i\}$ and $y_i = y$.
4. $(x'_1, x'_2, \cdots, x'_q) := \mathcal{B}(f(x_1), f(x_2), \cdots, f(x_q))$.
5. $f(x'_i) = y$ then output x'_i else $\perp$.

Figure 2.1: Construction of $\mathcal{A}(1^n, y)$

Note that: (1) $\mathcal{A}(1^n, y)$ iterates at most $T = \frac{4n^2}{\alpha_f(n)\mu_{\mathcal{B},g}(nq)}$ times each call is polynomial time. (2) $\mu_{\mathcal{B},g}(nq)$ is a non-negligible function. This implies that for infinite choices of n this value is greater than some noticeable function. Together these two facts imply that for infinite choices of n the running time of $\mathcal{A}$ is bounded by a polynomial function in n.

It remains to show that $\Pr_{x \xleftarrow{\$} \{0,1\}^n} [\mathcal{A}(1^n, f(x)) = \perp] < \alpha_f(n)$ for arbitrarily large n. A natural way to argue this is by showing that at least one execution of $\mathcal{B}$ should suffice for inverting $f(x)$. However, the technical challenge in proving this formally is that these calls to $\mathcal{B}$ aren't independent. Below we formalize this argument even when these calls aren't independent.

Lemma 2.1. *Let A be any an efficient algorithm such that $\Pr_{x,r}[A(x,r) = 1] \ge \epsilon$. Additionally, let $G = \{x \mid\ge \Pr_r[A(x,r) = 1] \ge \frac{\epsilon}{2}\}$. Then, we have $\Pr_x[x \in G] \ge \frac{\epsilon}{2}$.*

Proof. The proof of this lemma follows by a very simple counting argument. Let's start by assuming that $\Pr_x[x \in G] < \frac{\epsilon}{2}$. Next, observe that

$$\begin{aligned}
&\Pr_{x,r}[A(x,r) = 1] \\
&\quad = \Pr_x[x \in G] \cdot \Pr_{x,r}[A(x,r) = 1 \mid x \in G] \\
&\quad + \Pr_x[x \notin G] \cdot \Pr_{x,r}[A(x,r) = 1 \mid x \notin G] \\
&\quad < \frac{\epsilon}{2} \cdot 1 + 1 \cdot \frac{\epsilon}{2} \\
&\quad < \epsilon,
\end{aligned}$$

which is a contradiction. $\square$

Define the set S of "bad" x's, which are hard to invert:

$$S := \left\{ x \,\middle|\, \Pr_{\mathcal{B}}[\mathcal{A} \text{ inverts } f(x) \text{ in a single iteration}] \leq \frac{\alpha_f(n)\mu_{\mathcal{B},g}(nq)}{4n} \right\}.$$

We start by proving that the size of S is small. More formally,

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n}[x \in S] \leq \frac{\alpha_f(n)}{2}.$$

Assume, for the sake of contradiction, that $\Pr_{x \xleftarrow{\$} \{0,1\}^n}[x \in S] > \frac{\alpha_f(n)}{2}$. Then we have that:

$$\begin{aligned}
\mu_{\mathcal{B},g}(nq) = & \Pr_{(x_1,\cdots,x_q) \xleftarrow{\$} \{0,1\}^{nq}}[\mathcal{B}(1^{nq}, g(x_1,\cdots,x_q)) \in g^{-1}(g(x_1,\cdots,x_q))] \\
= & \Pr_{x_1,\cdots,x_q}[\mathcal{B}(1^{nq}, g(x_1,\cdots,x_q)) \in g^{-1}(g(x_1,\cdots,x_q)) \wedge \forall i : x_i \notin S] \\
& + \Pr_{x_1,\cdots,x_q}[\mathcal{B}(1^{nq}, g(x_1,\cdots,x_q)) \in g^{-1}(g(x_1,\cdots,x_q)) \wedge \exists i : x_i \in S] \\
\leq & \Pr_{x_1,\cdots,x_q}[\forall i : x_i \notin S] + \sum_{i=1}^{q} \Pr_{x_1,\cdots,x_q}[\mathcal{B}(1^{nq}, g(x_1,\cdots,x_q)) \in g^{-1}(g(x_1,\cdots,x_q)) \wedge x_i \in S] \\
\leq & \left(1 - \frac{\alpha_f(n)}{2}\right)^q + q \cdot \Pr_{x_1,\cdots,x_q,i}[\mathcal{B}(1^{nq}, g(x_1,\cdots,x_q)) \in g^{-1}(g(x_1,\cdots,x_q)) \wedge x_i \in S] \\
= & \left(1 - \frac{\alpha_f(n)}{2}\right)^{\frac{2n}{\alpha_f(n)}} + q \cdot \Pr_{x \xleftarrow{\$} \{0,1\}^n, \mathcal{B}}[\mathcal{A} \text{ inverts } f(x) \text{ in a single iteration} \wedge x \in S] \\
\leq & e^{-n} + q \cdot \Pr_x[x \in S] \cdot \Pr[\mathcal{A} \text{ inverts } f(x) \text{ in a single iteration} \mid x \in S] \\
\leq & e^{-n} + \frac{2n}{\alpha_f(n)} \cdot 1 \cdot \frac{\mu_{\mathcal{B},g}(nq) \cdot \alpha_f(n)}{4n} \\
\leq & e^{-n} + \frac{\mu_{\mathcal{B},g}(nq)}{2}.
\end{aligned}$$

Hence $\mu_{\mathcal{B},g}(nq) \leq 2e^{-n}$, contradicting with the fact that $\mu_{\mathcal{B},g}$ is non-negligible. Then we have

$$\begin{aligned}
& \Pr_{x \xleftarrow{\$} \{0,1\}^n}[\mathcal{A}(1^n, f(x)) = \perp] \\
& = \Pr_x[x \in S] + \Pr_x[x \notin S] \cdot \Pr[\mathcal{B} \text{ fails to invert } f(x) \text{ in every iteration} | x \notin S] \\
& \leq \frac{\alpha_f(n)}{2} + (\Pr[\mathcal{B} \text{ fails to invert } f(x) \text{ a single iteration} | x \notin S])^T \\
& \leq \frac{\alpha_f(n)}{2} + \left(1 - \frac{\mu_{\mathcal{A},g}(nq) \cdot \alpha_f(n)}{4n}\right)^T \\
& \leq \frac{\alpha_f(n)}{2} + e^{-n} \leq \alpha_f(n)
\end{aligned}$$

for sufficiently large n. This concludes the proof. □

Lemma 2.2. *Let A be any an efficient algorithm such that $\Pr_{x,r}[A(x_1, \ldots x_n, r) = 1] \geq \epsilon$. Additionally, let $G = \{x \mid \geq \Pr_{x_1,\ldots x_n,r}[A(x,r) = 1 \mid \exists i, x = x_i] \geq \frac{\epsilon}{2}\}$. Then, we have $\Pr_x[x \in G] \geq \frac{\epsilon}{2}$.*

Proof. The proof of this lemma follows by a very simple counting argument. Let's start by assuming that $\Pr_x[x \in G] < \frac{\epsilon}{2}$. Next, observe that

$$\begin{aligned}
& \Pr_{x,r}[A(x,r) = 1] \\
& = \Pr_x[x \in G] \cdot \Pr_{x,r}[A(x,r) = 1 \mid x \in G] \\
& + \Pr_x[x \notin G] \cdot \Pr_{x,r}[A(x,r) = 1 \mid x \notin G] \\
& < \frac{\epsilon}{2} \cdot 1 + 1 \cdot \frac{\epsilon}{2} \\
& < \epsilon,
\end{aligned}$$

which is a contradiction. □

2.4 *Levin's One-Way Function*

Theorem 2.2. *If there exists a one-way function, then there exists an explicit function f that is one-way (constructively).*

Lemma 2.3. *If there exists a one-way function computable in time n^c for a constant c, then there exists a one-way function computable in time n^2.*

Proof. Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function computable in time n^c. Construct $g : \{0,1\}^{n+n^c} \to \{0,1\}^{n+n^c}$ as follows:

$$g(x,y) = f(x)||y$$

where $x \in \{0,1\}^n, y \in \{0,1\}^{n^c}$. $g(x,y)$ takes time $2n^c$, which is linear in the input length.

We next show that $g(\cdot)$ is one-way. Assume for the purpose of contradiction that there exists an adversary $\mathcal{A}$ such that $\mu_{\mathcal{A},g}(n+n^c) = \Pr_{(x,y)\xleftarrow{\$}\{0,1\}^{n+n^c}}[\mathcal{A}(1^{n+n^c}, g(x,y)) \in g^{-1}(g(x,y))]$ is non-negligible. Then we use $\mathcal{A}$ to construct $\mathcal{B}$ such that $\mu_{\mathcal{B},f}(n) = \Pr_{x\xleftarrow{\$}\{0,1\}^n}[\mathcal{B}(1^n, f(x)) \in f^{-1}(f(x))]$ is also non-negligible.

$\mathcal{B}$ on input $z \in \{0,1\}^n$, samples $y \xleftarrow{\$} \{0,1\}^{n^c}$, and outputs the n higher-order bits of $\mathcal{A}(1^{n+n^c}, z||y)$. Then we have

$$\begin{aligned}
\mu_{\mathcal{B},g}(n) &= \Pr_{x\xleftarrow{\$}\{0,1\}^n, y\xleftarrow{\$}\{0,1\}^{n^c}}\left[\mathcal{A}(1^{n+n^c}, f(x)||y) \in f^{-1}(f(x))||\{0,1\}^{n^c}\right] \\
&\geq \Pr_{x,y}\left[\mathcal{A}(1^{n+n^c}, g(x,y)) \in f^{-1}(f(x))||y\right] \\
&= \Pr_{x,y}\left[\mathcal{A}(1^{n+n^c}, g(x,y)) \in g^{-1}(g(x,y))\right]
\end{aligned}$$

is non-negligible. □

of Theorem 2.2. We first construct a weak one-way function $h : \{0,1\}^n \to \{0,1\}^n$ as follows:

$$h(M,x) = \begin{cases} M||M(x) & \text{if } M(x) \text{ takes no more than } |x|^2 \text{ steps} \\ M||0 & \text{otherwise} \end{cases}$$

where $|M| = \log n, |x| = n - \log n$ (interpreting M as the code of a machine and x as its input). If h is weak one-way, then we can construct a one-way function from h as we discussed in Section 2.3.

It remains to show that if one-way functions exist, then h is a weak one-way function, with $\alpha_h(n) = \frac{1}{n^2}$. Assume for the purpose of contradiction that there exists an adversary $\mathcal{A}$ such that $\mu_{\mathcal{A},h}(n) = \Pr_{(M,x)\xleftarrow{\$}\{0,1\}^n}[\mathcal{A}(1^n, h(M,x)) \in h^{-1}(h(M,x))] \geq 1 - \frac{1}{n^2}$ for all sufficiently large n. By the existence of one-way functions and Lemma 2.3, there exists a one-way function $\tilde{M}$ that can be computed in time n^2. Let $\tilde{M}$ be the uniform machine that computes this one-way function. We will consider values n such that $n > 2^{|\tilde{M}|}$. In other words for these choices of n, $\tilde{M}$ can be described using $\log n$ bits. We construct $\mathcal{B}$ to invert $\tilde{M}$: on input y outputs the $(n - \log n)$ lower-order bits of $\mathcal{A}(1^n, \tilde{M}||y)$. Then

$$\begin{aligned}
\mu_{\mathcal{B},\tilde{M}}(n - \log n) &= \Pr_{x \stackrel{\$}{\leftarrow} \{0,1\}^{n-\log n}} \left[\mathcal{A}(1^n, \tilde{M}||\tilde{M}(x)) \in \{0,1\}^{\log n}||\tilde{M}^{-1}(\tilde{M}((x))\right] \\
&\geq \Pr_{x \stackrel{\$}{\leftarrow} \{0,1\}^{n-\log n}} \left[\mathcal{A}(1^n, \tilde{M}||\tilde{M}(x)) \in \tilde{M}||\tilde{M}^{-1}(\tilde{M}((x))\right].
\end{aligned}$$

Observe that for sufficiently large n it holds that

$$\begin{aligned}
1 - \frac{1}{n^2} \leq & \mu_{\mathcal{A},h}(n) \\
= & \Pr_{(M,x) \stackrel{\$}{\leftarrow} \{0,1\}^n} \left[\mathcal{A}(1^n, h(M,x)) \in h^{-1}(h(M,x))\right] \\
\leq & \Pr_M[M = \tilde{M}] \cdot \Pr_x \left[\mathcal{A}(1^n, \tilde{M}||\tilde{M}(x)) \in \tilde{M}||\tilde{M}^{-1}(\tilde{M}((x))\right] + \Pr_M[M \neq \tilde{M}] \\
\leq & \frac{1}{n} \cdot \mu_{\mathcal{B},\tilde{M}}(n - \log n) + \frac{n-1}{n}.
\end{aligned}$$

Hence $\mu_{\mathcal{B},\tilde{M}}(n - \log n) \geq \frac{n-1}{n}$ for sufficiently large n which is a contradiction. $\square$

Exercises

Exercise 2.1. *If $\mu(\cdot)$ and $\nu(\cdot)$ are negligible functions then show that $\mu(\cdot) \cdot \nu(\cdot)$ is a negligible function.*

Exercise 2.2. *If $\mu(\cdot)$ is a negligible function and $f(\cdot)$ is a function polynomial in its input then show that $\mu(f(\cdot))$[4] are negligible functions.*

[4] Assume that μ and f are such that $\mu(f(\cdot))$ takes inputs from $\mathbb{Z}^+$ and outputs values in $[0,1]$.

Exercise 2.3. *Prove that the existence of one-way functions implies $P \neq NP$.*

Exercise 2.4. *Prove that there is no one-way function $f : \{0,1\}^n \rightarrow \{0,1\}^{\lfloor \log_2 n \rfloor}$.*

Exercise 2.5. *Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ be any one-way function then is $f'(x) \stackrel{def}{=} f(x) \oplus x$ necessarily one-way?*

Exercise 2.6. *Prove or disprove: If $f : \{0,1\}^n \rightarrow \{0,1\}^n$ is a one-way function, then $g : \{0,1\}^n \rightarrow \{0,1\}^{n-\log n}$ is a one-way function, where $g(x)$ outputs the $n - \log n$ higher order bits of $f(x)$.*

Exercise 2.7. *Explain why the proof of Theorem 2.1 fails if the attacker $\mathcal{A}$ in Figure 2.1 sets $i = 1$ and not $i \stackrel{\$}{\leftarrow} \{1, 2, \cdots, q\}$.*

Exercise 2.8. *Given a (strong) one-way function construct a weak one-way function that is not a (strong) one-way function.*

Exercise 2.9. *Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ be a weak one-way permutation (a weak one way function that is a bijection). More formally, f is a PPT computable one-to-one function such that $\exists$ a constant $c > 0$ such that $\forall$ non-uniform PPT machine A and $\forall$ sufficiently large n we have that:*

$$\Pr_{x,A}[A(f(x)) \notin f^{-1}(f(x))] > \frac{1}{n^c}$$

Show that $g(x) = f^T(x)$ is not a strong one way permutation. Here f^T denotes the T times self composition of f and T is a polynomial in n.

Interesting follow up reading if interested: With some tweaks the function above can be made a strong one-way permutation using explicit constructions of expander graphs. See Section 2.6 in `http://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/part2N.ps`

3
Pseudorandomness

3.1 Distinguishability Between Two Distributions

Sanjam: add

3.2 Computational Indistinguishability

Defining indistinguishability between two distributions by a computationally bounded adversary turns out be tricky. In particular, It is tricky to define for a single pair of distributions because the length of the output of a random variable is a constant. Therefore, in order for "computationally bounded" adversaries to make sense, we have to work with infinite families of probability distributions.

Definition 3.1. *An* ensemble *of probability distributions is a sequence of random variables $\{X_n\}_{n \in \mathbb{N}}$. Two ensembles of probability distributions $\{X_n\}_n$ and $\{Y_n\}_n$ (which are* samplable *in time polynomial in n) are said to be* computationally indistinguishable *if for all (non-uniform) PPT machines $\mathcal{A}$, the quantities*

$$p(n) := \Pr[\mathcal{A}(1^n, X_n) = 1] = \sum_x \Pr[X_n = x] \Pr[\mathcal{A}(1^n, x) = 1]$$

and

$$q(n) := \Pr[\mathcal{A}(1^n, Y_n) = 1] = \sum_y \Pr[Y_n = y] \Pr[\mathcal{A}(1^n, y) = 1]$$

differ by a negligible amount; i.e. $|p(n) - q(n)|$ is negligible in n. This equivalence is denoted by

$$\{X_n\}_n \approx \{Y_n\}_n$$

We now prove some properties of computationally indistinguishable ensembles that will be useful later on.

Lemma 3.1 (Sunglass Lemma). *If $\{X_n\}_n \approx \{Y_n\}_n$ and P is a PPT machine, then*

$$\{P(X_n)\}_n \approx \{P(Y_n)\}_n$$

Proof. Consider an adversary $\mathcal{A}$ that can distinguish $\{P(X_n)\}_n$ from $\{P(Y_n)\}_n$ with non-negligible probability. Then the adversary $\mathcal{A} \circ P$ can distinguish $\{X_n\}_n$ from $\{Y_n\}_n$ with the same non-negligible probability. Since P and $\mathcal{A}$ are both PPT machines, the composition is also a PPT machine. This proves the contrapositive of the lemma. □

Lemma 3.2 (Hybrid Argument). *For a polynomial $t : \mathbb{Z}^+ \to \mathbb{Z}^+$ let the t-product of $\{Z_n\}_n$ be*

$$\{Z_n^{(1)}, Z_n^{(2)}, \ldots, Z_n^{(t(n))}\}_n$$

where the $Z_n^{(i)}$s are independent copies of Z_n. If

$$\{X_n\}_n \approx \{Y_n\}_n$$

then

$$\{X_n^{(1)}, \ldots, X_n^{(t)}\}_n \approx \{Y_n^{(1)}, \ldots, Y_n^{(t)}\}_n$$

as well.

Proof. Consider the set of tuple random variables

$$H_n^{(i,t)} = (Y_n^{(1)}, \ldots, Y_n^{(i)}, X_n^{(i+1)}, X_n^{(i+2)}, \ldots, X_n^{(t)})$$

for integers $0 \leq i \leq t$. Assume, for the sake of contradiction, that there is a PPT adversary $\mathcal{A}$ that can distinguish between $\{H_n^{(0,t)}\}_n$ and $\{H_n^{(t,t)}\}_n$ with non-negligible probability difference $r(n)$. Suppose that $\mathcal{A}$ returns 1 with probability ϵ_i when it runs on samples from $H_n^{(i,t)}$. By definition, $|\epsilon_t - \epsilon_0| \geq r(n)$. By the Triangle Inequality and the Pigeonhole Principle, there is some index k for which $|\epsilon_{k+1} - \epsilon_k| \geq r(n)/t$. However, using Sunglass Lemma, note that the computational indistinguishability of X_n and Y_n implies that $\{H_n^{(k,t)}\}_n$ and $\{H_n^{(k+1,t)}\}_n$ are computationally indistinguishable. This is a contradiction. □ □

3.3 Hard Core Bit

We start by asking the following question: Is it possible to concentrate the strength of a one-way function into one bit? In particular, given a one-way function f, does there exist one bit that can be computed efficiently from the input x, but is hard to compute given $f(x)$?

Definition 3.2 (Hard Core Bit)**.** *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function. $B : \{0,1\}^n \to \{0,1\}$ is a hard core bit of f if:*

- *B is computable by a polynomial time machine, and*
- *$\forall$ non-uniform PPT adversaries $\mathcal{A}$ we have that*

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n}[\mathcal{A}(1^n, f(x)) = B(x)] \leq \frac{1}{2} + \mathsf{negl}(n).$$

A simple example. Let f be a one-way function. Consider the one-way function $g(b,x) = 0||f(x)$ and a hard core bit $B(b,x) = b$. Intuitively, the value $g(b,x)$ does not reveal any information about the first bit b, thus no information about the value $B(b,x)$ can be ascertained. Hence $\mathcal{A}$ cannot predict the first bit with a non-negligible advantage than a random guess. However, we are more interested in the case where the hard core bit is hidden because of computational hardness and not information theoretic hardness.

Remark 3.1. *Given a one-way function f, we can construct another one-way function g with a hard core bit. However, we may not be able to find a hard core bit for f. In fact, it is an open question whether a hard core bit exists for every one-way function.*

Intuitively, if a function f is one-way, there should be a particular bit in the input x that is hard to compute given $f(x)$. But this is not true:

Claim 3.1. *If $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function, then there exists a one-way function $g : \{0,1\}^{n+\log n} \to \{0,1\}^{n+\log n}$ such that $\forall 1 \leq i \leq n + \log n$, $B_i(x) = x_i$ is not a hard core bit, where x_i is the i^{th} bit of x.*

Proof. Define $g : \{0,1\}^{n+\log(n)} \to \{0,1\}^{n+\log(n)}$ as follows.

$$g(x,y) = f(x_{\bar{y}})||x_y||y,$$

where $|x| = n$, $|y| = \log n$, $x_{\bar{y}}$ is all bits of x except the y^{th} bit, x_y is the y^{th} bit of x.

First, one can show that g is still a one-way function. (We leave this as an exercise!) We next show that B_i is not a hard core bit for $\forall 1 \leq i \leq n$ (clearly B_i is not a hard core bit for $n+1 \leq i \leq n + \log n$). Construct an adversary $\mathcal{A}_i(1^{n+\log n}, f(x_{\bar{y}})||x_y||y)$ that "breaks" B_i:

- If $y \neq i$ then output a random bit;
- Otherwise output x_y.

$$
\begin{aligned}
&\Pr_{x,y}[\mathcal{A}(1^{n+\log n}, g(x,y)) = B_i(x)] \\
=&\Pr_{x,y}[\mathcal{A}(1^{n+\log n}, f(x_{\bar{y}})||x_y||y) = x_i] \\
=&\frac{n-1}{n}\cdot\frac{1}{2}+\frac{1}{n}\cdot 1 = \frac{1}{2}+\frac{1}{2n}.
\end{aligned}
$$

Hence $\mathcal{A}_i$ can guess the output of B_i with greater than $\frac{1}{2} + \mathsf{negl}(n)$ probability. □

Application: Coin tossing over the phone. We next describe an application of hard core bits to coin tossing. Consider two parties trying to perform a coin tossing over the phone. In this setting the first party needs to declare its choice as the second one flips the coin. However, how can the first party trust the win/loss response from the second party? In particular, if the first party calls out "head" and then the second party can just lie that it was "tails." We can use hard core bit of a (one-to-one) one-way function to enable this applications.

Let f be a (one-to-one) one-way function and B be a hard core bit for f. Consider the following protocol:

- Party P_1 samples x from $\{0,1\}^n$ uniformly at random and sends y, where $y = f(x)$, to party P_2.
- P_2 sends back a random bit b sampled from $\{0,1\}$.
- P_1 sends back $(x, B(x))$ to P_2. P_2 aborts if $f(x) \neq y$.
- Both parties output $B(x) \oplus b$.

Note that P_2 cannot guess $B(x)$ with a non-negligible advantage than $1/2$ as he sends back his b. On the other hand, P_1 cannot flip the value $B(x)$ once it has sent $f(x)$ to P_2 because f is one-to-one.

3.4 Hard Core Bit of any One-Way Functions

We now show that a slight modification of every one-way function has a hard core bit. More formally,

Theorem 3.1. *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function. Define a function $g : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as follows:*

$$g(x,r) = f(x)||r,$$

where $|x| = |r| = n$. Then we have that g is one-way and that it has a hard core bit, namely $B(x,r) = \sum_{i=1}^{n} x_i r_i \mod 2$.

Remark 3.2. *If f is a (one-to-one) one-way function, then g is also a (one-to-one) one-way function with hard core bit $B(\cdot)$.*

Proof. We leave it as an exercise to show that g is a one-way function and below we will prove that the function $B(\cdot)$ describe a hard core bit of g. More specifically, we need to show that if there exists a non-uniform PPT $\mathcal{A}$ s.t. $\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)] \geq \frac{1}{2} + \epsilon(n)$, where ϵ is non-negligible, then there exists a non-uniform PPT $\mathcal{B}$ such that $\Pr_{x,r}[\mathcal{B}(1^{2n}, g(x,r)) \in g^{-1}(g(x,r))]$ is non-negligible. Below we use E to denote the event that $\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)$. We will provide our proof in a sequence of three steps of complexity: (1) the super simple case where we restrict to $\mathcal{A}$ such that $\Pr_{x,r}[E] = 1$, (2) the simple case where we restrict to $\mathcal{A}$ such that $\Pr_{x,r}[E] \geq \frac{3}{4} + \epsilon(n)$, and finally (3) the general case with $\Pr_{x,r}[E] \geq \frac{1}{2} + \epsilon(n)$.

Super simple case. Suppose that $\mathcal{A}$ breaks the B with perfect accuracy:

$$\Pr_{x,r}[E] = 1.$$

We now construct $\mathcal{B}$ that inverts g with perfect accuracy. Let e^i be an n-bit string $0 \cdots 010 \cdots 0$, where only the i-th bit is 1, the rest are all 0. On input $f(x)||R$, $\mathcal{B}$ does the following:

for $i = 1$ **to** n **do**
  $x'_i \leftarrow \mathcal{A}(1^{2n}, f(x)||e^i)$
end for
return $x'_1 \cdots x'_n||R$

Observe that $B(x, e^i) = \sum_{j=1}^{n} x_j e^i_j = x_i$. Therefore, the probability that $\mathcal{B}$ inverts a single bit successfully is,

$$\Pr_x\left[\mathcal{A}(1^{2n}, f(x)||e^i) = x_i\right] = \Pr_x\left[\mathcal{A}(1^{2n}, f(x)||e^i) = B(x, e^i)\right] = 1.$$

Hence $\Pr_{x,r}[\mathcal{B}(1^{2n}, g(x,r)) = (x,r)] = 1$.

Simple case. Next moving on to the following more demanding case.

$$\Pr_{x,r}[E] \geq \frac{3}{4} + \epsilon(n),$$

where ϵ is non-negligible. Just like the super simple case, we describe our algorithm of $\mathcal{B}$ for inverting g. On input $f(x)||R$, $\mathcal{B}$ proceeds as follows:

for $i = 1$ **to** n **do**
  for $t = 1$ **to** $T = \frac{n}{2\epsilon(n)^2}$ **do**
    $r \xleftarrow{\$} \{0,1\}^n$
    $x_i^t \leftarrow \mathcal{A}(f(x)||r) \oplus \mathcal{A}(f(x)||r + e^i)$
  end for
  $x'_i \leftarrow$ the majority of $\{x_i^1, \cdots, x_i^T\}$
end for

return $x'_1 \cdots x'_n || R$

Correctness of $\mathcal{B}$ given that $\mathcal{A}$ calls output the correct answer follows by observing that $B(x,r) \oplus B(x, r \oplus e^i) = x_i$:

$$\begin{aligned}
&B(x,r) \oplus B(x, r \oplus e^i) \\
=&\sum_j x_j r_j + \sum_j x_j (r_j \oplus e^i_j) \mod 2 \\
=&\sum_{j \neq i} (x_j r_j + x_j r_j) + x_i r_i + x_i (r_i + 1) \mod 2 \\
=&x_i.
\end{aligned}$$

The key technical challenge in proving that $\mathcal{B}$ inverts g with non-negligible probability arises from the fact that the calls to $\mathcal{A}$ made during one execution of $\mathcal{B}$ are not independent. In particular, all calls to $\mathcal{A}$ share the same x and the class $\mathcal{A}(f(x)||r)$ and $\mathcal{A}(f(x)||r + e^i)$ use correlated randomness as well. We solve the first issue by showing that exists a large choices of values of x for which $\mathcal{A}$ still works with large probability. The later issue of lack of independent of $\mathcal{A}(f(x)||r)$ and $\mathcal{A}(f(x)||r + e^i)$ will be solved using a union bound. Formally, define the set G of "good" x's, which are easy for $\mathcal{A}$ to predict:

$$G := \left\{ x \,\middle|\, \Pr_r [E] \geq \frac{3}{4} + \frac{\epsilon(n)}{2} \right\}.$$

We start by proving that the size of G is not small. More formally we claim that,

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n} [x \in G] \geq \frac{\epsilon(n)}{2}.$$

Assume, that $\Pr_{x \xleftarrow{\$} \{0,1\}^n} [x \in G] < \frac{\epsilon(n)}{2}$. Then we have the following contradiction:

$$\begin{aligned}
\frac{3}{4} + \epsilon(n) \leq &\Pr_{x,r}[E] \\
=&\Pr_x[x \in G] \Pr_r[E | x \in G] + \Pr_x[x \notin G] \Pr_r[E | x \notin G] \\
<&\frac{\epsilon(n)}{2} \cdot 1 + 1 \cdot \left(\frac{3}{4} + \frac{\epsilon(n)}{2} \right) = \frac{3}{4} + \epsilon(n).
\end{aligned}$$

For and fixed $x \in G$:

$$\begin{aligned}
&\Pr_r \left[\mathcal{A}(f(x), r) \oplus \mathcal{A}(f(x), r + e^i) = x_i \right] \\
=&\Pr_r [\text{Both } \mathcal{A}\text{'s are correct}] + \Pr_r [\text{Both } \mathcal{A}\text{'s are wrong}] \\
\geq&\Pr_r [\text{Both } \mathcal{A}\text{'s are correct}] \\
\geq&1 - 2 \cdot \Pr_r [\text{Either } \mathcal{A} \text{ is correct}] \\
\geq&1 - 2 \left(\frac{1}{4} - \frac{\epsilon(n)}{2} \right) = \frac{1}{2} + \epsilon(n).
\end{aligned}$$

Let Y_i^t be the indicator random variable that $x_i^t = x_i$ (namely, $Y_i^t = 1$ with probability $\Pr[x_i^t = x_i]$ and $Y_i^t = 0$ otherwise). Note that $Y_i^1, \cdots, Y_i^T$ are independent and identical random variables, and for all $t \in \{1, \ldots, T\}$ we have that $\Pr[Y_i^t = 1] = \Pr[x_i^t = x_i] \geq \frac{1}{2} + \epsilon(n)$. Next we argue that majority of $x_i^1, \ldots x_i^T$ coincides with x_i with high probability.

$$\begin{aligned}
\Pr[x_i' \neq x_i] =& \Pr\left[\sum_{t=1}^{T} Y_i^t \leq \frac{T}{2}\right] \\
=& \Pr\left[\sum_{t=1}^{T} Y_i^t - \left(\frac{1}{2} + \epsilon(n)\right) T \leq \frac{T}{2} - \left(\frac{1}{2} + \epsilon(n)\right) T\right] \\
\leq& \Pr\left[\left|\sum_{t=1}^{T} Y_i^t - \left(\frac{1}{2} + \epsilon(n)\right) T\right| \geq \epsilon(n) T\right]
\end{aligned}$$

Let $X_1, \cdots, X_m$ be i.i.d. random variables taking values 0 or 1. Let $\Pr[X_i = 1] = p$.

By Chebyshev's Inequality, $\Pr\left[\left|\sum X_i - pm\right| \geq \delta m\right] \leq \dfrac{1}{4\delta^2 m}$.

$$\leq \frac{1}{4\epsilon(n)^2 T} = \frac{1}{2n}.$$

Then, completing the argument, we have

$$\begin{aligned}
& \Pr_{x,r}[\mathcal{B}(1^{2n}, g(x,r)) = (x,r)] \\
\geq& \Pr_{x}[x \in G] \Pr[x_1' = x_1, \cdots x_n' = x_n | x \in G] \\
\geq& \frac{\epsilon(n)}{2} \cdot \left(1 - \sum_{i=1}^{n} \Pr[x_i' \neq x_i | x \in G]\right) \\
\geq& \frac{\epsilon(n)}{2} \cdot \left(1 - n \cdot \frac{1}{2n}\right) = \frac{\epsilon(n)}{4}.
\end{aligned}$$

Real Case. Now, we describe the final case where $\Pr_{x,r}[E] \geq \frac{1}{2} + \epsilon(n)$, where $\epsilon(\cdot)$ is a non-negligible function. The key technical challenge in this case is that we cannot make two related calls to $\mathcal{A}$ as was done in the simple case above. However, just using one call to $\mathcal{A}$ seems insufficient. The key idea is to just guess one of those values. Very surprisingly this idea along with careful analysis magically works out. Just like the previous two case we start by describing the algorithm $\mathcal{B}$. On input $f(x)||R$, $\mathcal{B}$ proceeds as follows:

$T = \frac{2n}{\epsilon(n)^2}$
for $\ell = 1$ **to** $\log T$ **do**
  $s_\ell \xleftarrow{\$} \{0,1\}^n$
  $b_\ell \xleftarrow{\$} \{0,1\}$
end for
for $i = 1$ **to** n **do**

for all $L \subseteq \{1, 2, \cdots, \log T\}$ **do**
  $S_L := \bigoplus_{j \in L} s_j$
  $B_L := \bigoplus_{j \in L} b_j$
  $x_i^L \leftarrow B_L \oplus \mathcal{A}(f(x)||S_L + e^i)$
end for
$x_i' \leftarrow$ the majority of $\{x_i^{\emptyset}, \cdots, x_i^{[\log T]}\}$
end for
return $x_1' \cdots x_n'||R$

The idea is the following. Let b_ℓ guess the value of $B(x, s_\ell)$, and with probability $\frac{1}{T}$ all the b_ℓ's are correct. In that case, it is easy to see that $B_L = B(x, S_L)$ for every L. If we follow the same argument as above, then it remains to bound the probability that $\mathcal{A}(f(x)||S_L + e^i) = B(x, S_L + e^i)$. However there is a subtle issue. Now the events $Y_i^{\emptyset}, \cdots, Y_i^{[\log T]}$ are not independent any more. But we can still show that they are pairwise independent, and the Chebyshev's Inequality still holds. Now we give the formal proof.

Just as in the simple case, we define the set G as

$$G := \left\{ x \middle| \Pr_r [E] \geq \frac{1}{2} + \frac{\epsilon(n)}{2} \right\},$$

and with an identical argument we obtain that

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n} [x \in G] \geq \frac{\epsilon(n)}{2}.$$

Correctness of $\mathcal{B}$ follows from the fact in case $b_\ell = B(x, s_\ell)$ for every $\ell \in [\log T]$ then $\forall L \subseteq [\log T]$, it holds that (we use the notation $(s)_k$ to denote the k^{th} bit of s)

$$B(x, S_L) = \sum_{k=1}^{n} x_k \left(\bigoplus_{j \in L} s_j \right)_k = \sum_{k=1}^{n} x_k \sum_{j \in L} (s_j)_k = \sum_{j \in L} \sum_{k=1}^{n} x_k (s_j)_k = \sum_{j \in L} B(x, s_j) = \sum_{j \in L} b_j = B_L.$$

Next given that $b_\ell = B(x, s_\ell), \forall \ell \in [\log T]$ and $x \in G$ we bound the probability,

$$\begin{aligned}
\Pr_r \left[B_L \oplus \mathcal{A}(f(x)||S_L + e^i) = x_i \right] &= \Pr_r \left[B(x, S_L) \oplus \mathcal{A}(f(x)||S_L + e^i) = x_i \right] \\
&= \Pr_r \left[\mathcal{A}(f(x)||S_L + e^i) = B(x, S_L + e^i) \right] \\
&\geq \frac{1}{2} + \frac{\epsilon(n)}{2}.
\end{aligned}$$

For $b_\ell = B(x, s_\ell), \forall \ell \in [\log T]$ and $x \in G$, let Y_i^L be the indicator random variable that $x_i^L = x_i$. Notice that $Y_i^{\emptyset}, \cdots, Y_i^{[\log T]}$ are pairwise independent and $\Pr[Y_i^L = 1] = \Pr[x_i^L = x_i] \geq \frac{1}{2} + \frac{\epsilon(n)}{2}$.

$$\begin{aligned}
\Pr[x_i' \neq x_i] &= \Pr\left[\sum_{L \subseteq [\log T]} Y_i^L \leq \frac{T}{2}\right] \\
&= \Pr\left[\sum_{L \subseteq [\log T]} Y_i^L - \left(\frac{1}{2} + \frac{\epsilon(n)}{2}\right) T \leq \frac{T}{2} - \left(\frac{1}{2} + \frac{\epsilon(n)}{2}\right) T\right] \\
&\leq \Pr\left[\left|\sum_{L \subseteq [\log T]} Y_i^L - \left(\frac{1}{2} + \frac{\epsilon(n)}{2}\right) T\right| \geq \frac{\epsilon(n)}{2} T\right] \\
&\quad \text{(By Theorem 3.2)} \\
&\leq \frac{1}{4\left(\frac{\epsilon(n)}{2}\right)^2 T} = \frac{1}{2n}.
\end{aligned}$$

Then, completing the proof, we have that

$$\begin{aligned}
&\Pr_{x,r}[\mathcal{B}(1^{2n}, g(x,r)) = (x,r)] \\
&\geq \Pr\left[\forall \ell \in [\log T], b_\ell = B(x, s_\ell)\right] \\
&\quad \cdot \Pr_x[x \in G] \Pr[x_1' = x_1, \cdots x_n' = x_n | \forall \ell \in [\log T], b_\ell = B(x, s_\ell), x \in G] \\
&\geq \frac{1}{T} \cdot \frac{\epsilon(n)}{2} \cdot \left(1 - \sum_{i=1}^{n} \Pr[x_i' \neq x_i | \forall \ell \in [\log T], b_\ell = B(x, s_\ell), x \in G]\right) \\
&\geq \frac{\epsilon(n)^2}{2n} \cdot \frac{\epsilon(n)}{2} \cdot \left(1 - n \cdot \frac{1}{2n}\right) = \frac{\epsilon(n)^3}{8n}.
\end{aligned}$$

□

Pairwise Independence and Chebyshev's Inequality. Here, for the sake of completeness, we prove the Chebyshev's Inequality.

Definition 3.3 (Pairwise Independence). *A collection of random variables $\{X_1, \cdots, X_m\}$ is said to be* pairwise independent *if for every pair of random variables $(X_i, X_j), i \neq j$ and every pair of values (v_i, v_j), it holds that*

$$\Pr[X_i = v_i, X_j = v_j] = \Pr[X_i = v_i] \Pr[X_j = v_j].$$

Theorem 3.2 (Chebyshev's Inequality). *Let $X_1, \ldots, X_m$ be pairwise independent and identically distributed binary random variables. In particular, for every $i \in [m]$, $\Pr[X_i = 1] = p$ for some $p \in [0,1]$ and $\Pr[X_i = 0] = 1 - p$. Then it holds that*

$$\Pr\left[\left|\sum_{i=1}^{m} X_i - pm\right| \geq \delta m\right] \leq \frac{1}{4\delta^2 m}.$$

Proof. Let $Y = \sum_i X_i$. Then

$$\begin{aligned}
\Pr\left[\left|\sum_{i=1}^{m} X_i - pm\right| > \delta m\right] &= \Pr\left[\left(\sum_{i=1}^{m} X_i - pm\right)^2 > \delta^2 m^2\right] \\
&\leq \frac{\mathbb{E}\left[|Y - pm|^2\right]}{\delta^2 m^2} \\
&= \frac{\mathrm{Var}(Y)}{\delta^2 m^2}
\end{aligned}$$

Observe that

$$\begin{aligned}
\mathrm{Var}(Y) &= \mathbb{E}\left[Y^2\right] - (\mathbb{E}[Y])^2 \\
&= \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\mathbb{E}\left[X_i X_j\right] - \mathbb{E}\left[X_i\right] \mathbb{E}\left[X_j\right]\right)
\end{aligned}$$

By pairwise independence, for $i \neq j$, $\mathbb{E}\left[X_i X_j\right] = \mathbb{E}$

$$\begin{aligned}
&= \sum_{i=1}^{m} \mathbb{E}\left[X_i^2\right] - \mathbb{E}\left[X_i\right]^2 \\
&= mp(1-p).
\end{aligned}$$

Hence

$$\Pr\left[\left|\sum_{i=1}^{m} X_i - pm\right| \geq \delta m\right] \leq \frac{mp(1-p)}{\delta^2 m^2} \leq \frac{1}{\delta^2 m}.$$

□

Bibliography

Mihir Bellare. A note on negligible functions. *Journal of Cryptology*, 15 (4):271–284, September 2002. DOI: 10.1007/s00145-002-0116-x.