The Goldreich-Levin Theorem

Arpon Basu (Roll No: 200050013)

April 29, 2023

Contents

1 Some Definitions

We recap some basic definitions of cryptography before stating and proving the Goldreich-Levin theorem.

Definition 1 (One-Way Function)**.** *A family of functions $f_n : \{0,1\}^n \mapsto \{0,1\}^{k(n)}$ are called one-way functions if they are computable in polynomial time and for every non-uniform* PPT *adversary $\mathcal{A}$,*

$$\Pr_{x \leftarrow \{0,1\}^n}(f_n(\mathcal{A}(f_n(x))) = f_n(x)) = \mathit{negligible}(n)$$

where $\mathit{negligible}(n)$ is a function which decays super-polynomially with n.

Definition 2 (Hard-Core Predicate)**.** *A predicate $h : \{0,1\}^* \to \{0,1\}$ is called a hard-core predicate for a one-way function $f : \{0,1\}^n \mapsto \{0,1\}^{k(n)}$ if h is computable in polynomial time and for every non-uniform* PPT *adversary $\mathcal{A}$*

$$\Pr_{x \leftarrow \{0,1\}^n}(\mathcal{A}(1^n, f(x)) = h(x)) = \frac{1}{2} + \mathit{negligible}(n)$$

We extensively referred to [1], [2], and [3] for our report.

2 Theorem Statement

The Goldreich-Levin theorem goes as follows:

Theorem 2.1 (Goldreich-Levin Theorem)**.** *Let f be a one-way function with domain $\{0,1\}^n$. Note that for any $r \in \{0,1\}^n$, $g(x,r) := (f(x), r)$ is a one-way function too. Then $h(x,r) := \langle x, r\rangle$ is a hard-core predicate for g, where $\langle x, r\rangle$ denotes the dot product of x and r (in $\mathbb{F}_2$).*

3 The Proof

We proceed via contradiction: Consider a PPT adversary which can guess the hardcore bit with non-negligible probability over $\frac{1}{2}$. We shall construct a PPT adversary which can invert f with non-negligible probability.
However establishing the theorem requires some lemmata, which we shall now prove.

Lemma 3.1. *Let $\mathcal{A}$ be any PPT adversary, let $\delta > 0$. Define*

$$G_{\mathcal{A},\delta} := \left\{ x : \Pr_{r \leftarrow \{0,1\}^n}(\mathcal{A}(f(x), r) = \langle x, r\rangle) \geq \frac{1+\delta}{2} \right\}$$

If $\Pr_{x,r\leftarrow\{0,1\}^n}(\mathcal{A}(f(x), r) = \langle x, r\rangle) \geq \frac{1}{2} + \delta$, then $\Pr_{x\leftarrow\{0,1\}^n}(x \in G_{\mathcal{A},\delta}) \geq \frac{\delta}{2}$.

Proof. Note that

$$\Pr_{x,r\leftarrow\{0,1\}^n}(\mathcal{A}(f(x), r) = \langle x, r\rangle) = \Pr_{x,r\leftarrow\{0,1\}^n}(\mathcal{A}(f(x), r) = \langle x, r\rangle | x \in G_{\mathcal{A},\delta}) \Pr_{x\leftarrow\{0,1\}^n}(x \in G_{\mathcal{A},\delta})$$

$$+ \Pr_{x,r\leftarrow\{0,1\}^n}(\mathcal{A}(f(x), r) = \langle x, r\rangle | x \notin G_{\mathcal{A},\delta}) \Pr_{x\leftarrow\{0,1\}^n}(x \notin G_{\mathcal{A},\delta})$$

$$\leq 1 \cdot \Pr_{x\leftarrow\{0,1\}^n}(x \in G_{\mathcal{A},\delta}) + \frac{1+\delta}{2} \cdot 1$$

Since $\Pr_{x,r\leftarrow\{0,1\}^n}(\mathcal{A}(f(x), r) = \langle x, r\rangle) \geq \frac{1}{2} + \delta$, we get our desired result. □

Lemma 3.2. *Let $X_1, X_2, \ldots, X_{m'}$ be pairwise independent Bernoulli random variables with parameter p. Define $X := \sum_{i=1}^{m'} X_i$. Then*

$$\Pr(|X - \mathbb{E}[X]| \geq m'\delta) \leq \frac{1}{4m'\delta^2}$$

Proof. Denote by μ the value of $\mathbb{E}[X]$.
Note that

$$\mathrm{Var}(X) = \mathbb{E}[(X-\mu)^2] = \mathbb{E}[X^2 - 2\mu X + \mu^2]$$

$$= \mathbb{E}[X^2] - 2\mu\mathbb{E}[X] + \mu^2 = \mathbb{E}\left[\sum_{i=1}^{m'} X_i^2 + 2\sum_{1\leq i<j\leq m'} X_i X_j\right] - 2\mu\mathbb{E}[X] + \mu^2$$

$$= \sum_{i=1}^{m'} \mathbb{E}[X_i^2] + 2 \sum_{1 \le i < j \le m'} \mathbb{E}[X_i X_j] - 2\mu^2 + \mu^2$$

Since X_i, X_j are pairwise independent for $i \neq j$, $\mathbb{E}[X_i X_j] = \mathbb{E}[X_i]\mathbb{E}[X_j] = p^2$. Moreover, $\mathbb{E}[X_i^2] = p$. Consequently

$$\mathrm{Var}(X) = \sum_{i=1}^{m'} p + 2 \sum_{1 \le i < j \le m'} p^2 - \mu^2 = m'p(1-p)$$

where the last equality follows since $\mu = m'p$.
The desired result then follows by invoking Chebyshev's inequality and noting that $p(1-p) \le \frac{1}{4}$ for every $p \in [0, 1]$. $\square$

Now, let $\delta = \text{non-negligible}(n) > 0$ be the advantage of our adversary $\mathcal{A}$ in calculating the hardcore bit, ie:- $\Pr_{x,r \leftarrow \{0,1\}^n}(\mathcal{A}(f(x), r) = \langle x, r \rangle) \ge \frac{1}{2} + \delta$. Set $m := \lceil \frac{2n}{\delta^2} \rceil, k := 1 + \lceil \log_2(m) \rceil$. Uniformly choose k random vectors $t_1, t_2, \ldots, t_k$ from $\{0,1\}^k$. Now, let $S \subseteq \{1, 2, \ldots, k\} =: [k]$ be any non-empty set. Then we define r_S as $r_S := \sum_{i \in S} t_i$. This way we can generate $2^k - 1 = m' \ge m$ random vectors. Note that all the vectors r_S are themselves distributed uniformly in $\{0,1\}^n$ since a linear combination of uniform random vectors from $\{0,1\}^n$ is itself a uniform random vector [1].
Note that for any two sets $S_1 \neq S_2$, r_{S_1}, r_{S_2} are independent. Consequently, all our m' random vectors are pairwise independent.
Now assume we already know the correct values of $\langle x, t_i \rangle$ for every $i \in [k]$. Then we know the values $\langle x, r_S \rangle$ for every $S \subseteq [k]$, since $\langle x, r_S \rangle = \langle x, \sum_{i \in S} t_i \rangle = \sum_{i \in S} \langle x, t_i \rangle$.
Let e_i be the i^{th} unit vector of $\{0,1\}^n$. For any $S \subseteq [k]$, since r_S are uniformly random, we get that $r_S \oplus e_i$ is uniformly random too. Moreover, note that $\langle x, r_S \oplus e_i \rangle - \langle x, r_S \rangle = \langle x, e_i \rangle = x_i$.
Consequently, for every $S \subseteq [k]$, calculate the value of $\mathcal{A}(f(x), r_S \oplus e_i) - \langle x, r_S \rangle$, where $\mathcal{A}$ is the adversary calculating the hardcore bit, obtain m' votes for the value of x_i, and take the *majority vote* of these values [2].
Let ξ_S be the Bernoulli random variable denoting the probability distribution of $\mathcal{A}(f(x), r_S \oplus e_i)$ correctly calculating $\langle x, r_S \oplus e_i \rangle$. If $x \in G_{\mathcal{A},\delta}$, then the parameter of ξ_S is at least $\frac{1+\delta}{2}$, by the definition given in Lemma 3.1.
Consequently, the expected number of correct answers in the m' votes for the value of x_i is at least $\frac{m'(1+\delta)}{2}$, and thus if the majority vote turns up the wrong answer, that implies a deviation from the mean of more than $\frac{m'\delta}{2}$. By Lemma 3.2, the probability of this happening is at most $\frac{1}{m'\delta^2} \le \frac{1}{m\delta^2} \le \frac{1}{2n}$.
Consequently, the probability that any bit is calculated wrongly is at most $\frac{1}{2n}$, which implies, by the union bound, that the probability that x is determined wrongly is at most $\frac{1}{2n} \cdot n = \frac{1}{2}$. Note that x is simply determined by

[1] this can be seen through induction
[2] since $m' = 2^k - 1$ is an odd number, a tie is not possible

a concatenation of the bits x_i for $i \in [n]$.
Consequently, we managed to invert $f(x)$ with probability $\geq \frac{1}{2} \cdot \Pr(x \in G_{\mathcal{A},\delta}) \geq \frac{\delta}{4}$. However since δ is not negligible, neither is $\frac{\delta}{4}$, which implies that with non-negligible probability we can invert $f(x)$, violating the assumption that it was a one-way function.
We still have to deal with one small catch: We assumed that we know $\langle x, t_i \rangle$ for every $i \in [n]$. But obviously, that is not true *a priori*. We deal with this as follows: We run the aforementioned algorithm for all $2^k = m' + 1 = \text{poly}(n)$ possible values of $(\langle x, t_i \rangle)_{i \in [k]}$. Every time, we end up with a possible value of x, whose correctness we test for by checking if applying $f(x)$ is the correct answer. Since we know that for the correct values of $(\langle x, t_i \rangle)_{i \in [k]}$, we obtain the correct value of x with probability at least $\frac{1}{2}$, we can consequently conclude that we will get the correct answer with probability at least $\frac{1}{2}$ by the end of all the 2^k iterations.
The above step blows up our runtime by 2^k, but since 2^k is polynomial in n, our algorithm remains polynomial time, and thus our overall construction of a PPT adversary continues to hold.

4 Connection with local list decoding of Hadamard Codes

The construction used to generate $2^k - 1$ pairwise independent random vectors is very similar to the concept of local list decoding for Hadamard codes: For any $x \in \{0,1\}^n$, the Hadamard encoding of x, denoted $\text{Had}(x)$ is defined as $\text{Had}(x) := (\langle x, y \rangle)_{y \in \{0,1\}^n} \in \{0,1\}^{2^n}$.
In the context of the Goldreich-Levin theorem, the reason why the Hadamard code is so important is because it is $(q, \delta, \varepsilon) = \left(2, \frac{1}{4}, 0\right)$-locally decodable: What this means is that if y is a noisy/corrupted version of $\text{Had}(x)$ such that $\|y - \text{Had}(x)\|_1 \leq \delta \cdot n = \frac{n}{4}$, then sampling just 2-bits of y allows us to recover any bit of x with probability atleast $\frac{1}{2} + \varepsilon = \frac{1}{2}$.
The recovery technique of the above local decoding is exactly same as how we obtained the i^{th} bit of x in the proof of the Goldreich-Levin theorem: For a uniformly random $r \in \{0,1\}^n$, sample the bit of y corresponding to $\langle x, r \rangle =: y_r$. Then x_i can be computed as $y_r \oplus y_{r \oplus e_i}$, where e_i is the standard i^{th} basis vector, and moreover this calculation is correct with probability $\geq \frac{1}{2} + \frac{1}{2} - 2\delta = \frac{1}{2}$, as claimed.
Thus the proof of Goldreich-Levin theorem is quite commonly referred to in literature as being equivalent to the list decoding of the Hadamard code.

5 Applications of the Goldreich-Levin Theorem

One of the most immediate and useful applications of this theorem is to construct *pseudo-random generators* (PRGs): Indeed, let $f : \{0,1\}^n \mapsto \{0,1\}^n$ be a one-way permutation. Then $g(x, r) = f(x)||r||\langle x, r\rangle$ is a pseudo-random generator [3]. Indeed, through this construction, the Goldreich-Levin theorem lays the foundation for constructing a large class of PRGs.
This construction can be easily extended to a stretch of polynomial length. Indeed,

Theorem 5.1. *If f is a one-way permutation, then*

$$g_N(x, r) := r||\langle f^N(x), r\rangle||\langle f^{(N-1)}(x), r\rangle|| \ldots ||\langle f(x), r\rangle||\langle x, r\rangle$$

is a PRG for any $N \sim poly(n)$, and f^k denotes the k-fold composition of f.

Proof. We know that pseudorandomness is equivalent to a next-bit prediction by Yao's theorem.
Now assume for the sake of contradiction that g is not a PRG: Then there would exist $i \in [N]$ and a PPT adversary $\mathcal{A}$ such that

$$\Pr(\mathcal{A}(r||\langle f^N(x), r\rangle||\langle f^{N-1}(x), r\rangle|| \ldots ||\langle f^{i+1}(x), r\rangle) = \langle f^i(x), r\rangle) = \frac{1}{2} + \varepsilon$$

We describe a PPT adversary $\mathcal{B}$ such that given $(f(z), r)$, $\mathcal{B}$ tells us the value of $\langle z, r\rangle$ with non-negligible probability, thus violating the Goldreich-Levin theorem.
$\mathcal{B}$ chooses an $i \in [N]$ randomly. Consider $x \in \{0,1\}^n$ such that $f^i(x) = z$ [4]. Note that for $\ell \geq 1$, $\mathcal{B}$ can efficiently calculate $f^{i+\ell}(x) = f^{\ell-1}(f(z))$. Consequently, $\mathcal{B}$ can, in polynomial time, generate the string $r||\langle f^N(x), r\rangle|| \ldots ||\langle f^{i+1}(x), r\rangle$ on it's own and feed it to $\mathcal{A}$ as an input, which would then return to $\mathcal{B}$ the value of $\langle z, r\rangle$ with non-negligible probability, allowing $\mathcal{B}$ to violate the Goldreich-Levin theorem. □

References

[1] Arora and Barak. *Computational Complexity: A Modern Approach.* 2007. URL: https://theory.cs.princeton.edu/complexity/book.pdf.

[2] Omkant Pandey. *Hard Core Predicates.* 2017. URL: https://www3.cs.stonybrook.edu/~omkant/L05-short.pdf.

[3] Omkant Pandey. *Proof of GL Theorem.* 2017. URL: https://www3.cs.stonybrook.edu/~omkant/L06.pdf.

[3]this can be proved through the equivalence of the definitions of pseudo-randomness and next-bit unpredictability

[4]Such a x must necessarily exist since the composition of two permutations is also a permutation, and consequently every element in our co-domain has a (unique) pre-image