CSC 2426: Fundamentals of Cryptography **Fall 2023**

Lecture 3: Proof of Goldreich-Levin Theorem

Instructor: Akshayaram Srinivasan *Scribe: Vladyslav Nekriach*

Date: September 25 2023

In the last class, we argued that if $\{h_n\}_{n\in\mathbb{N}}$ is a hard-core predicate for $\{f_n\}_{n\in\mathbb{N}}$, then $\{f_n\}_{n\in\mathbb{N}}$ is a one-way function. This shows that one-way functions are necessary for the existence of hard-core predicates. In this lecture, we will show that they are sufficient.

Theorem 3.1 (Goldreich-Levin [GL89]) *If one-way functions (OWFs) exist, then there* $\exists\{g_n, h_n\}_{n\in\mathbb{N}}$ *s.t. and* $\{h_n\}_{n\in\mathbb{N}}$ *is a hard-core predicate for* $\{g_n\}_{n\in\mathbb{N}}$.

3.1 Proof of Theorem 3.1

Let $f = \{f_n\}_{n\in\mathbb{N}}$ be a one-way function where

$$f_n : \{0,1\}^{k(n)} \longrightarrow \{0,1\}^{m(n)}, \forall n \in \mathbb{N}$$

Let's define another family of functions $g = \{g_n\}_{n\in\mathbb{N}}$ where

$$g_n = \{0,1\}^{2k(n)} \rightarrow \{0,1\}^{k(n)+m(n)}, \forall n \in \mathbb{N}$$

where input to g_n is split in 2 parts x and r, each consisting of $k(n)$ bits. We use $(x_1, ..., x_{k(n)})$ to denote the bit representation of x and $(r_1, \ldots, r_{k(n)})$ denote the bit representation of r.

We define g_n in the following way:

$$g_n(x, r) = f_n(x) \mid\mid r, \forall n \in \mathbb{N}, \text{ where } \mid\mid \text{ represents concatenation operation}$$

We also define $h = \{h_n\}_{n\in\mathbb{N}}$ in the following way:

$$h_n(x, r) = \langle x, r\rangle, \forall n \in \mathbb{N}, \text{ where } \langle x, r\rangle \text{ represents } (\sum_{i=1}^{k(n)} x_i \cdot r_i) \bmod 2$$

We will now prove that h is a hard-core predicate for g. A necessary condition for this to happen is that g is one-way. Let's verify that this is indeed the case.

Claim 3.2 *g is one-way if f is one-way.*

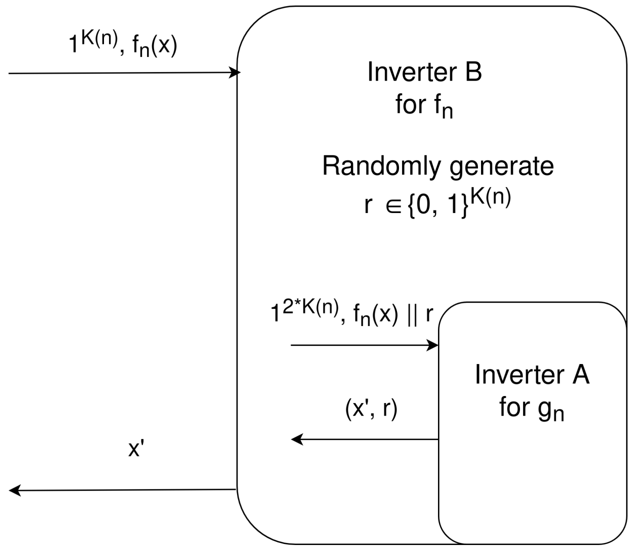

Figure 3.1: Construction of inverter for f_n to prove the one-wayness of g_n.

Proof: Suppose by contradiction that g is not a OWF. This means that there exists a non-uniform PPT $\mathcal{A}$ that can invert g with non-negligible probability. We will use $\mathcal{A}$ to design an inverter $\mathcal{B}$ for f.

The construction of $\mathcal{B}$ is given in Figure 3.1. On input $(1^{k(n)}, f_n(x))$, $\mathcal{B}$ samples r randomly. It passes $(1^{2k(n)}, f_n(x)\|r)$ to $\mathcal{A}$. $\mathcal{A}$ outputs (x', r) and $\mathcal{B}$ outputs x'.

It can be easily verified that the probability that $\mathcal{B}$ inverts f is at least the probability that $\mathcal{A}$ inverts g, which is assumed to be non-negligible. This contradicts the one-wayness of f. ■

We just verified that g is a one-way function. But this doesn't still prove that h is a hard-core predicate for g. Assume for the sake of contradiction that h is not a hard-core predicate. This means that there exists a nuPPT $\mathcal{A}$ and polynomial p such that for infinitely many n, we have:

$$\Pr_{(x,r)\leftarrow\{0,1\}^{2k(n)}}[\mathcal{A}(1^{2k(n)}, f_n(x)\|r) = h_n(x,r)] \geq \frac{1}{2} + \frac{1}{p(n)}$$

We will use $\mathcal{A}$ to design an inverter $\mathcal{B}$ for f.

3.1.1 Easy Case

Let's first consider the case where $\mathcal{A}$ predicts h with probability 1:

$$\Pr_{(x,r)\leftarrow\{0,1\}^{2k(n)}}[\mathcal{A}(1^{2k(n)}, f(x)||r) = h_n(x,r)] = 1$$

We will design $\mathcal{B}$ as follows. Let $e_i = (0,0,0,...,0,1,0,...,0)$ be a vector of length $k(n)$ that has 1 in the i-th position. If we pass a value $f_n(x)\|e_i$ to $\mathcal{A}$, $\mathcal{A}$ would always correctly compute the value of i-th bit of x correctly due to the fact that $\mathcal{A}$ is always correct. We can pass $e_1, \ldots, e_{k(n)}$ through $\mathcal{A}$ to compute each bit of x. This inverter always succeeds and this contradicts the one-wayness of f.

3.1.2 Non-Trivial Case

Let's now weaken the requirements that $\mathcal{A}$ predicts h. Specifically, let us consider the case where

$$\Pr_{(x,r)\leftarrow\{0,1\}^{2k(n)}}[\mathcal{A}(1^{2k(n)}, f_n(x)\|r) = h_n(x,r)] \geq \frac{3}{4} + \frac{1}{p(n)}$$

for infinitely many n.

The previous approach does not work anymore due to the fact that inverter $\mathcal{A}$ might fail on some of the instances of $f_n(x)$ and $r = e_i$, giving false information about x, therefore, x will be inverted incorrectly.

To solve this we define a set Good_n, which is:

$$\mathsf{Good}_n = \{x \in \{0,1\}^{k(n)} | \Pr_{r\leftarrow\{0,1\}^{k(n)}}[\mathcal{A}(1^{2k(n)}, f_n(x)\|r) = h_n(x,r)] \geq \frac{3}{4} + \frac{1}{2p(n)}\}$$

Claim 3.3 $\Pr_{x\leftarrow\{0,1\}^{k(n)}}[x \in \mathsf{Good}_n] \geq \frac{1}{2p(n)}$

Proof:

$$\begin{aligned}
\frac{3}{4} + \frac{1}{p(n)} &\leq \Pr_{x,r}[\mathcal{A} \text{ predicts } h_n] \\
&= \Pr_x[x \in \mathsf{Good}_n] \cdot \Pr_r[\mathcal{A} \text{ predicts } h_n \,|\, x \in \mathsf{Good}_n] \\
&\qquad + \Pr_x[x \notin \mathsf{Good}_n] \cdot \Pr_r[\mathcal{A} \text{ predicts } h_n \,|\, x \notin \mathsf{Good}_n] \\
&\leq \Pr_x[x \in \mathsf{Good}_n] + \Pr_r[\mathcal{A} \text{ predicts } h_n \,|\, x \notin \mathsf{Good}_n] \\
&\leq \Pr_x[x \in \mathsf{Good}_n] + \frac{3}{4} + \frac{1}{2p(n)}
\end{aligned}$$

This shows that $\Pr_x[x \in \mathsf{Good}_n] \geq \frac{1}{2p(n)}$. ■

We now try to mimic the procedure from the easy case of the theorem. For that, we use the fact that $\langle x, r\rangle \oplus \langle x, r \oplus e_i\rangle = \langle x, r \oplus r \oplus e_i\rangle = x_i$. Note that if r is randomly generated, $r \oplus e_i$ is also random, despite being correlated to r.

This property of inner product allows us to try to probabilistically invert i-th bit of x by trying multiple r values, for each of them performing 2 queries $\langle x, r\rangle, \langle x, r \oplus e_i\rangle$ to the inverter, taking XOR of the answers and doing a majority vote afterwards.

If $x \in \mathsf{Good}_n$, each query with randomly chosen r errs with probability $\frac{1}{4} - \frac{1}{2p(n)}$. Due to the union bound, probability that both queries are correct is $1 - (\frac{1}{4} - \frac{1}{2p(n)}) \cdot 2 = \frac{1}{2} + \frac{1}{p(n)} > \frac{1}{2}$.

We can model each attempt with a random variable $Z_j, j = 1...m$ (m is yet to be estimated) that takes the value 1 iff x_i obtained through the above process is correct. Therefore, $\Pr[Z_j = 1] \geq \frac{1}{2} + \frac{1}{p(n)}$. Let $Z = \sum_{i=1}^{m} Z_i$.

$$\mathbb{E}[Z] = (\frac{1}{2} + \frac{1}{p(n)}) \cdot m = \frac{m}{2} + \frac{m}{p(n)}$$

$$\Pr[Z \leq \frac{m}{2}] \leq \Pr[|Z - E(Z)| \geq \frac{m}{p(n)}] = \leq 2e^{\frac{-2(\frac{m}{p(n)})^2}{m}}$$

For $m = n \cdot p(n)^2$, the probability of being wrong on i-th bit is $\leq 2e^{-2n}$.

Therefore, the probability that we don't err in computing any x_i :

$$\Pr[\text{inverter succeeds} | x \in \mathsf{Good}_n] \geq 1 - 2ne^{-2n}$$

Hence,

$$\begin{aligned}
\Pr[\text{inverter succeeds}] &\geq \Pr[x \in \mathsf{Good}_n] \cdot \Pr[\text{inverter succeeds} | x \in \mathsf{Good}_n] \\
&\geq \frac{1}{2p(n)} \cdot (1 - 2ne^{-2n})
\end{aligned}$$

where RHS is non-negligible.

References

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 25–32. ACM, 1989.