CS276: Cryptography September 3, 2015

Universal OWF and Hardcore predicates

Instructor: Alessandro Chiesa *Scribe: Akshayaram Srinivasan*

1 Overview

In last lecture we discussed about the hardness amplification lemma for One-Way Function (OWF). In particular , we saw how to convert a weak one way function to a strong one. In this lecture we will look at the notion of an Universal OWF, hardcore predicates for OWF and discuss the Goldreich-Levin construction of a hardcore predicate for a One-Way Permutation (OWP).

2 Universal OWF

Universal OWF theorem constructs a specific OWF under the assumption that OWF exists. On a philosophical note, the theorem says that even if the candidate constructions of OWFs like RSA, Discrete Log etc are broken there exists a function which is one-way if $\mathsf{P} \neq \mathsf{NP}$. More formally,

Theorem 1 *[Lev87] If one way functions exist then there exists a specific function f^* which is one way.*

Proof: To prove this theorem, we will first show that if OWFs exist then there is there is a OWF which can be evaluated in time quadratic in its input length. Using this fact we will then construct a specific function which is one-way.

Lemma 2 *If $\{f_k\}_k$ is a family of one-way functions then there exists another family of functions $\{g_k\}_k$ such that $\{g_k\}_k$ is one-way and for all $k \in \mathbb{N}$, g_k can be evaluated in time $(n_g(k))^2$*

Proof: The proof of this lemma uses a technique called as *Padding* which has has its roots in complexity theory. Let $f_k : \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}$ be one-way. From the property of one-way functions (efficient evaluation) there exists a specific polynomial $p(\cdot)$ such that for all $k \in \mathbb{N}$, f_k can be evaluated in time $p(n(k))$. We now define a function $g_k : \{0,1\}^{p(n(k))} \to \{0,1\}^{m(k)+p(n(k))-n(k)}$ such that $g_k(x||w) = f(x)||w$ where $|x| = n(k)$ and $|w| = p(n(k)) - n(k)$. We first claim that $\{g_k\}_k$ is one-way.

Claim 3 *If f_k[1] is one-way then so is g_k.*

Proof: Assume for the sake of contradiction that g_k is not one-way. Then there exists an adversary A such that A inverts $g_k(x)$ for a random x in the domain of g_k with non-negligible probability. We will be using A to invert f_k.

[1] For the ease of notation we will be considering f_k instead of the function family $\{f_k\}_k$

$I(y)$

- Sample $w \xleftarrow{\$} \{0,1\}^{p(n(k))-n(k)}$
- $x||w \leftarrow A(y||w, 1^{n(k)})$.
- Output x

It is easy to see that the I inverts y with the same probability as the inversion probability of A which is non-negligible from our assumption. This is a contradiction to the fact that f_k is one-way.

□

Now lets analyze the evaluation time of g_k. We can parse the input into $x||w$ in time $p(n(k))^2$ [2]. Evaluating f_k takes time $p(n(k))$ and hence the total time for evaluating g_k is bounded by $p(n(k))^2$. □

Lets now construct the universal OWF f^*. Let $M_1, M_2, \cdots$, be an enumeration of the Turing machines such that $M_i(|x|)$ runs in time $\mathsf{poly}(i, |x|)$. Note that such an enumeration can be done by an uniform machine given the size of the alphabet. We define $f^*(x)$ as :

$$f^*(x) = M_1^{\leq |x|^2}(x)||M_2^{\leq |x|^2}(x)|| \cdots ||M_{|x|}^{\leq |x|^2}(x)$$

where $M_i^{\leq |x|^2}(x)$ denotes running the machine M_i on input x for at most $|x|^2$ steps. We first observe that f^* can be computed in time polynomial in the length of $|x|$. The enumeration of the machines takes time $O(|x|)$ as we are interested in $|x|$ machines and running each machine takes $|x|^2$ time. Hence, f^* can be computed in time $O(|x|^3)$. Now, we show that f^* is one-way. Since g_k can be computed by a poly-time machine there exists an index N such that M_N computes g_k. For all $|x| > N$, $f^*(x)$ computes $g_k(x)$ in the index N. Since g_k is one way, it is also easy to see that f^* is one-way. □

3 Hardcore predicates

Lets define the notion of a hardcore predicate for a one-way function.

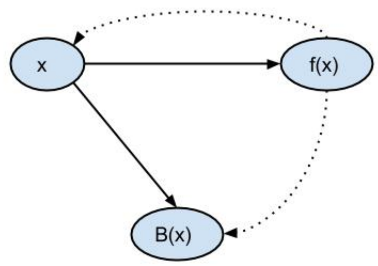
Definition 4 *$B_k : \{0,1\}^{n(k)} \to \{0,1\}$ is a hardcore predicate for a one-way function $f_k : \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}$ if*

- *B_k is efficiently computable.*
- *It is "hard" to compute $B_k(x)$ given k and $f_k(x)$. Formally, for all non-uniform PPT adversaries A,*

$$Pr\left[b = B_k(x) \middle| \begin{array}{l} x \xleftarrow{\$} \{0,1\}^{n(k)} \\ y \leftarrow f_k(x) \\ b \leftarrow A(1^k, f_k(x)) \end{array}\right] \leq 1/2 + negl(k)$$

[2]An one tape Turing machine might take quadratic time to parse the input.

Pictorially we could represent the notion of one-way functions and hardcore bits as follows:



Lets see if there exits a specific index $i \in [n(k)]$ such that $B_k(x) = x_i$ is hardcore for a one-way function.

Claim 5 *There is a one-way function family $\{g_k\}_k$ such that for all $i \in [n(k)]$, $B_k^i(x) = x_i$ is not hardcore for g_k.*

Proof: Let $f_k : \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}$ be a one-way function. Lets now construct a function family $g_k : \{0,1\}^{n(k)+1+\log(n(k)+1)} \to \{0,1\}^{m(k)+1+\log(n(k)+1)}$ where

$$g_k(z) = g_k(x||j) = f_k(x_{-j})||x_j||j$$

The explanation for the above equation is that g_k first parses the input into $n(k)+1$ bit x and $\log(n(k)+1)$ bit j. It then applies f on all bits of x except j^{th} bit. That is, $x_{-j} = x_1 \cdots x_{j-1}x_{j+1} \cdots x_{n(k)+1}$. It then outputs $f(x_{-j})||x_j||j$.

It is easy to see that g_k can be computed in polynomial time and is one-way given that f is one way (It follows a similar argument as in Claim 3). We now show that for all $i \in [n(k)]$, $B_k^i(x) = x_i$ is not a hardcore predicate for g_k. To prove this, we construct an adversary A_i which will predict $B_k^i(x)$ with non-negligible advantage.

$$A_i(Y) = A_i(y||x_j||j) = \begin{Bmatrix} x_j \ , \ j = i \\ b \stackrel{\$}{\leftarrow} \{0,1\} \ , \ j \neq i \end{Bmatrix}$$

We now claim that the $Pr[b = B_k^i(x)]$ is $\frac{1}{2} + \frac{1}{(2(n(k)+1))}$. This follows directly from the observation that $j = i$, happens with probability $\frac{1}{n(k)+1}$ since $j \in \{0,1\}^{\log(n(k)+1)}$ and is sampled uniformly at random for the generating the challenge. □

A natural question to ask is whether there exists a hardcore bit B_k for an one-way function f_k. It is still an open problem!

The next question we ask is given a one-way function f_k can we construct a one-way function g_k and predicate B_k such that B_k is hardcore for g_k. This is trivial to achieve. Consider $g_k(b||x) = 0||f_k(x)$ and $B_k(b||x) = b$. One can easily verify that the hardcore bit is information theoretically hidden.

Lets consider the following question.

Given a one-way permutation f_k, does there exists a one-way permutation g_k and a predicate B_k such that B_k is hardcore for g_k?.

The answer to the above question was given by Goldreich and Levin in [GL89].

Theorem 6 *[GL89] If f_k is a OWP then there exists a OWP g_k and a predicate B_k such that B_k is a hardcore predicate for g_k*

Proof: We will first construct a OWP g_k from a OWP f_k and then define the hardcore predicate for g_k [3].

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a OWP. We define $g : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as

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

It is easy to see that since f is a permutation so is g. The one-wayness of g follows from a similar argument as in Claim 3. Now lets define a predicate for g and then show that the predicate is indeed hardcore. The predicate we are going to consider is:

$$B(x||r) = < x, r > \mod 2$$

B can be computed efficiently (in polynomial time).

Lemma 7 *$B(x||r)$ is hard to compute with non-negligible advantage greater than $1/2$ given $g(x||r)$*

Proof: Lets assume for the sake of contradiction that $B(x||r)$ is can be computed with non-negligible advantage greater than $1/2$ given $g(x||r)$. Then there exists an adversary A and a polynomial $p(\cdot)$ such that for infinitely many k's:

$$\delta_A = Pr\left[b = (< x, r > \mod 2) \middle| \begin{array}{l} x, r \xleftarrow{\$} \{0,1\}^n \\ y \leftarrow f(x) \\ b \leftarrow A(1^k, y||r) \end{array} \right] > 1/2 + 1/p(k)$$

We will now consider an inverter for f using A. We will motivate the intuition for the proof by considering the following scenarios.

[3]For the ease of notation we will be ignoring the subscript k in f_k, g_k and B_k

- *Warmup 1:* $\delta_A = 1$: Lets define e^i to be a bit string of length n such that the i^{th} position has a 1 and the rest are 0. The inverter $I(y)$ for f works as follows: for every $i \in [n]$, compute $b_i \leftarrow A(y||e^i)$ and finally output $b_1 \cdots b_n$. Lets see why the Inverter works. Since $\delta_A = 1$, A is able to correctly output the hardcore bit for every x, r. In particular, it should output the hardcore it for x, e_i for all $i \in [n]$. Since $< x, e_i > \mod 2 = x_i$ the Inverter is able to correctly output x such that $f(x) = y$.

- *Warmup II:* $\delta_A > 3/4 + 1/p(n)$. We first observe that we cannot use the same trick as before because we cannot bound the probability that A correctly outputs the hardcore bit for every e^i. We will now make use of the fact that inner product is a bi-linear function. We observe that $< x, e_i > = < x, r > \oplus < x, r \oplus e_i >$.

  We say that $x \in \{0,1\}^n$ is *good* if

$$\Pr_{r,A}[A(f(x)||r) = < x, r >] \geq \frac{3}{4} + \frac{1}{2p(n)}$$

  where the probability also includes the random coin tosses made by A.

  If x is *good* then, we would like to estimate the probability that

$$\begin{aligned}
\Pr_{r,A}\begin{bmatrix} A(f(x)||r) = < x, r > \\ \wedge A(f(x)||r \oplus e_i) = < \\ x, r \oplus e_i > \end{bmatrix} &= 1 - \Pr_{r,A}[A(f(x)||r) \neq < x, r > \vee A(f(x)||r \oplus e_i) \neq < x, r \oplus e_i >] \\
&\geq 1 - (\Pr_{r,A}[A(f(x)||r) \neq < x, r >] + \Pr_{r,A}[A(f(x)||r \oplus e_i) \neq < x, r >]) \\
&\geq 1 - (\frac{1}{4} - \frac{1}{2p(n)}) - (\frac{1}{4} - \frac{1}{2p(n)}) \\
&= (\frac{1}{2} + \frac{1}{p(n)})
\end{aligned}$$

  The first inequality follows from the previous equation as a result of union bound and the second inequality follows from the definition of x is *good*.

  We are ready to describe the inverter $I(y)$ that inverts the one-way challenge y.

$I(y)$

- for $i = 1, \cdots, n$
  - for $j = 1 \cdots, m = poly(p(n))$
    - $r \xleftarrow{\$} \{0,1\}^n$
    - $c_{i,j} \leftarrow A(y||r) \oplus A(y||r \oplus e_i)$
  - $b_i \leftarrow Majority(c_{i1}, \cdots, c_{im})$
- Output $b_1 \cdots b_n$

  By a simple application of Chernoff bound we get the success probability that I correctly computes b_i to be at least $1 - \frac{1}{n2^n}$. The probability that we don't error in any of the $i's$ is at least $1 - \frac{1}{2^n}$ from union bound.

  We will now prove that the number of *good* $x's$ is at least $\frac{2^n}{2p(n)}$. We will now show that this will complete the analysis for this case.

We have seen above that

$$Pr[I \text{ inverts } f(x) | x \text{ is } good] \geq 1 - \frac{1}{2^n}$$

$$\begin{aligned} Pr[I \text{ inverts } f(x)] &\geq Pr[I \text{ inverts } f(x) | x \text{ is } good] Pr[x \text{ is } good] \\ &\geq (1 - \frac{1}{2^n}) \frac{1}{2p(n)} \\ &\geq \frac{1}{3p(n)} \end{aligned}$$

which is non-negligible.

Claim 8 $|good| \geq \frac{2^n}{2p(n)}$

Proof: Assume for the sake of contradiction that $|good| < \frac{2^n}{2p(n)}$.

$$\begin{aligned} Pr[A(f(x)||r) =< x, r >] &= Pr[A(f(x)||r) =< x, r > | x \text{ is not } good] Pr[x \text{ is not } good] \\ &+ Pr[x \text{ is } good].Pr[A(f(x)||r) =< x, r > | x \text{ is } good] \\ &\leq Pr[A(f(x)||r) =< x, r > | x \text{ is not } good] + Pr[x \text{ is } good] \\ &\leq \frac{3}{4} + \frac{1}{2p(n)} + \frac{1}{2p(n)} \\ &= \frac{3}{4} + \frac{1}{p(n)} \end{aligned}$$

which is a contradiction to the assumption that $\delta_A \geq 3/4 + 1/p(n)$. □

- *Warmup III:* $\delta_A \geq 1/2 + 1/(p(n))$ and an additional assumption which we will describe later. Like in the previous case, we will define $x \in \{0,1\}^n$ to be *ok* if

$$\Pr_{r,A}[A(f(x)||r) =< x, r >] \geq \frac{1}{2} + \frac{1}{2p(n)}$$

By an exact same argument as in Claim 8 we can prove that the number of *ok* x's is at least $\frac{2^n}{2p(n)}$. But now we cannot prove that A will succeed in outputting the hardcore predicate for both $f(x)||r$ and $f(x)||r \oplus e_i$ with non-negligible advantage greater than 1/2 if $x \in ok$. Therefore, we will make an additional assumption that there exists an oracle θ which on input y draws m independent samples $r_1, \cdots, r_m$ uniformly from $\{0,1\}^n$ and outputs $(r_1, z_1), \cdots (r_m, z_m)$ where for each $i \in [m]$, $z_i =< f^{-1}(y), r_i >$. Now the inverter just has to query A on input $y||r_j \oplus e_i$ for each $j \in [m]$ and take the majority. The inverter $I(Y)$ works as follows:

$I(y)$

- for $i = 1, \cdots, n$
  - $(r_1, z_1) \cdots (r_m, z_m) \leftarrow \theta(y)$
  - for $j = 1 \cdots, m = poly(p(n))$
    - $c_{i,j} \leftarrow z_j \oplus A(y||r_j \oplus e_i)$
  - $b_i \leftarrow Majority(c_{i1}, \cdots, c_{im})$
- Output $b_1 \cdots b_n$

The analysis of the success probability of I is similar to the above case.

- $\delta_A \geq 1/2 + 1/p(n)$ In this we relax the requirement that such a θ exists.

  We first make the observation that for the majority of c_{ij}'s to be correct with probability $1 - \frac{1}{n.n}$ it is enough that all the $r_i's$ are pairwise independent and not totally independent (by Chebychev's tail bounds for pairwise independent variables). Now we will try to simulate the effect of θ.

$\theta(y)$

- Sample $r_1, \cdots, r_{\log m} \xleftarrow{\$} \{0,1\}^m$
- Sample $z_1, \cdots, z_{\log m} \xleftarrow{\$} \{0,1\}$
- For $S \subseteq [\log m]$
  * Compute $r_S = \oplus_{i \in S}(r_i)$
  * Compute $z_S = \oplus_{i \in S}(z_s)$

It is easy to observe that r_S's are pairwise independent. The final observation is that $z_1, \cdots, z_{\log m}$ are all correct with probability $1/m$. By linearity of inner product with probability $1/m$ all the z_S's are correct and hence $\theta(y)$ is correct with probability $1/m$. The analysis is similar to the above case but we also get a factor of $1/m$ in the success probability of I.

□

Thus, we can conclude from Lemma 7 and the observation that B is efficiently computable that B is a hardcore predicate for g □

References

[GL89] Oded Goldreich and Leonid A Levin, *A hard-core predicate for all one-way functions*, Proceedings of the twenty-first annual ACM symposium on Theory of computing, ACM, 1989, pp. 25–32.

[Lev87] Leonid A Levin, *One way functions and pseudorandom generators*, Combinatorica **7** (1987), no. 4, 357–363.