**CSE 594 : Modern Cryptography** 02/07/17

# Lecture 5: Hard Core Predicates

*Instructor: Omkant Pandey* *Scribe: Gangabarani Balakrishnan, Ravikumar Rajendran*

## 1 Last Class

In the Last class, We saw how to convert weak one way functions into strong one way function and we saw an example of this with the multiplication function $f_x$. And we also saw the hardness amplification which converts somewhat hard problem into a really hard problem.

## 2 What do OWFs hide?

The concept of OWFs is simple and concise, But OWFs often not very useful by themselves. The OWFs only guarantees that $f(x)$ hides $x$ but nothing more. For example, If you take predicate $a$ which is some information about $x$, if the $f$ is not an identity function or close to some identity function, then for any non trivial $a$, $f$ may not hide that $a$.

Then we may ask the question "Is there a non-trivial function $f$ even if it is one bit that the OWFs hide?" the Answer is the Hardcore predicate.

## 3 Hardcore Predicate

A hard core predicate for a OWF $f$ is a function over its inputs $x$. The output of the function is a single bit (hardcore bit). The output bit can be easily computed given $x$. But the output bit is hard to compute given $f(x)$. The reason this bit is called hardcore bit is because it is guaranteed hard to compute information about $x$. Learning the hardcore bit of $x$ given the $f(x)$ is as hard as inverting the function $f(x)$ and learning x.

The function $f$ may leak many bits of x but it does not leak the hard-core bit. "Hard to compute" for a single bit means that one cannot guess the value of the hardcore bit with probability better than 1/2 (plus a noticeable value). Hard-core bit cannot be efficiently "learned" or "predicted" or "computed" with probability $> 1/2 + \mu(|x|)$.

**Definition 1** *A predicate $h : \{0,1\}^* \to \{0,1\}$ is a hard-core predicate for if $h$ is efficiently computable given $x$ and there exists a negligible function $\nu$ s.t. for every non-uniform PPT adversary $A$ and $\forall n \in \mathbb{N}$:*

$$Pr\big[x \leftarrow \{0,1\}^n : A(1^n, f(x)) = h(x)\big] \leq 1/2 + \nu(n)$$

### 3.1 Construction

Is it possible to construct hard core predicates for general OWFs f? The answer is yes. Lets define $< x, r >$ to be the inner product function mod 2.

$$< x, r >= \Big(\sum_i x_i r_i\Big)$$

It is the same as taking xor ($\oplus$) of a random subset of bits of $x$.

**Theorem 1** *Theorem (Goldreich-Levin)*
*Let $f$ be a OWF (OWP). Define function*

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

*where $|x| = |r|$. Then $g$ is a OWF (OWP) and*

$$h(x, r) =< x, r >$$

*is a hard-core predicate for $g$.*

**Remark 1** *The theorem is not for $f$, but for a different function, $g$. The Theorem is still useful. Consider the function $g'$.*

$$g'(1x) = g'(0x) = f(x).$$

*In the above equation, its clear to see that the first bit of function $g$ is hard core for $g$ and it works even if $f$ is not one way!. The problem with the above is that it looses information about its input. This is not good for applications. It explains nothing about the inherent hardness of $f$ Function $g$ in the GL theorem* statistically *does not loose any information that $f$ does not about its input. And the hard core bit for $g$ is easy to guess if $f$ is not one-way.*

# 4 Proof by reduction

We will show that if a non-uniform PPT adversary $\mathcal{A}$, given $(f(x), r)$, can compute $h(x, r)$ with significantly better probability than 1/2, then there exists a non-uniform PPT adversary $\mathcal{B}$ that inverts $f(x)$.

## 4.1 Main Challenge

Adversary $\mathcal{A}$ of the hard core predicate function $h$ outputs only 1-bit. But, for the purpose of this proof, we need to build an inverting function $\mathcal{B}$ for OWF $f$ that outputs all n-bits of the input $x$.

## 4.2 Warmup Proof - 1

**Assumption 1** *Given OWF (OWP) $g(x) = (f(x), r)$, adversary $\mathcal{A}$ always - with probability 1 - outputs $h(x, r)$ correctly*

**Building Inverter $\mathcal{B}$**: Since adversary $\mathcal{A}$ always computes $h(x, r)$ correctly, we can construct $(f(x), r)$ such that $r$ has its $i^{th}$ bit set to 1 and all other bits are set to 0. Thus we obtain the bit $x_i$ as below.

**Proof.**

$$Compute : x_i^* \leftarrow \mathcal{A}(f(x), e_i) \; for \; every \; i \in [n] \; where,$$
$$e_i = (\underbrace{0, ....., 0,}_{\text{i-1 bits}} 1, 0, ....., 0)$$

$$Output : x^* = x_1^*, x_2^*.....x_n^*$$

■

### 4.3 Warmup Proof - 2

**Assumption 1** *Given OWF (OWP)* $g(x) = (f(x), r)$*, adversary* $\mathcal{A}$ *outputs* $h(x, r)$ *with probability* $3/4 + \epsilon(n)$

**Main Problem**: Adversary may detect and ignore improper inputs - One example for improper input could be $e_i$ in the previous case

**Building Inverter** $\mathcal{B}$: Here, we split each query into two queries such that each query looks random individually thus not giving the attacker any opportunity to identify it as an improper input.

1. Let the random queries be $a := \mathcal{A}(f(x), e_i \oplus r)$ and $b := \mathcal{A}(f(x), r)$ for $r \leftarrow \{0,1\}^n$
2. Compute $c := a \oplus b$ as a guess for $x_i^*$
3. Repeat *step* 2 many times to get value of $c$ agreed by majority for $x_i$
4. Output $x^* = x_1^*, x_2^*.....x_n^*$

**Proof.**

1. If both $a$ and $b$ are correct, then $c = x_i$ because,

$$\begin{aligned} c &= a \oplus b \\ &= \langle x, e_i \oplus r_i \rangle \oplus \langle x, r \rangle \\ &= x.(r + e_i) + x.r mod 2 \\ &= x.e_i \\ &= x_i \end{aligned}$$

2. **Claim**: $c = x_i$ with probability $1/2 + 2\epsilon$
   **Proof**: By union bound, the probability for $\mathcal{A}$ being wrong about either $a$ or $b$ is atmost:

$$\begin{aligned} Prob &= (1/4 - \epsilon(n)) + (1/4 - \epsilon(n)) \\ &= 1/2 - 2\epsilon \end{aligned}$$

   So, both $a$ and $b$ can be correct with probability $\geq 1/2 + 2\epsilon$ which applies to $c$ as well.

3. By **Chernoff Bound**, if we repeat computation of $c$ $2n/\epsilon(n)$ times, majority of $c$ will be correct $x_i^*$ with probability $1 - e^{-n}$.

■

### 4.4 More examples of OWF

1. **Discrete Log**: Compute $x \in G$ from $(g, y, p)$ where $g$ generates a group $G$, the size of the group $p = |G|$ is a prime, and $x$ solves the equation $y = g^x$ such that $y$ is also in the group G.
2. **RSA Problem**: Compute deciphered value $d$ from public key pair $(e, N)$−$(exponent,\ large\, semi\, prime)$ such that $e.d \equiv 1\ mod\ \phi(N)$ where $\phi(N) = |\mathbb{Z}_{\mathbb{N}}^*|$ (Euler's totient function)
3. **Quadratic Residuosity**: Given 2 integers $a$ and $N$, $a$ is said to be quadratic residue modulo $N$ if there exists an integer $x$ such that, $a \equiv x^2\ (mod\, N)$
4. **More**: There are more examples for OWFs from lattices and LWE(Learning With Errors) problem. But, such "hardness assumptions" are few and rare.

### 4.5 Remarks

1. From the above OWFs, we get a collection of OWFs instead of a single OWF. However, collection of OWFs imply a single OWF as well - (will be discussed in the later lectures).
2. Special hard-core predicates and specific structures of these functions can yield more than 1 hard-core bit. For general OWFs, GL can be extended to yield $log\, n$ hard-core bits.
3. Although OWFs are necessary for many cryptographic applications, they are not sufficient for things like key-exchange and public-key encryption.
4. **Universal One-way functions**: If there exists any OWF (even if explicitly not known), Levin shows that an explicit OWF can be constructed based on the existence of the former.
5. Based on the current evidence, it is unlikely that OWFs whose hardness can be reduced to the validity of $P \neq NP$ exist.

### 4.6 Markov's Inequality

**Statement**: If $X$ is a non-negative random variable and $r > 0$, then

$$\mathbb{P}[X \geq r] \leq \frac{\mathbb{E}[X]}{r} \tag{1}$$

**Proof.**
We know that, the expectation of $X$ is,

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{x \in\, supported(X)} x.P(X = x) \\
&\leq 0 + r.\sum_{x>r} P_r[X = x] \\
&\leq r.P_r[X > r] \Rightarrow (1)
\end{aligned}$$

■

### 4.7 Chebyshev's Inequality

**Statement**: If $X$ is a non-negative random variable and $A > 0$, then

$$\mathbb{P}[|X - E[X]| > A] \leq \frac{Var(X)}{A^2} \tag{2}$$

Let,

$$Y = (X - E[X])^2$$

We know that,

$$Var[X] = E[(X - E[X])^2]$$
$$Var[X] = E[Y]$$

From Markov's inequality,

$$\Pr[Y > A^2] \leq \frac{E[Y]}{A^2}$$
$$\leq \frac{Var[X]}{A^2} \Rightarrow (2)$$

■

# 5 Chernoff's Bound

For independent random variables, The previous bounds are not tight enough. That is why we use chernoff's Bound.

let $X_1, X_2, ..., X_n$ be Random variables. And

$$\Pr[X_i = 1] = P_i; X = \Sigma X_i$$

**Linearity of Expectation:**

$$E[X_1 + X_2] = E[X_1] + E[X_2]$$

where $X_1, X_2$ are a random variables and $E[X]$ represents the Expectation value of random variable $X$. In case, $X_1$ and $X_2$ are independent random variables,

$$E[X_1.X_2] = E[X_1].E[X_2]$$

$$E[X_i] = P_i$$
$$\mu = E[X]$$
$$\mu = E[\sum_i X_i]$$
$$\mu = \sum_i E[X_i]$$

$$\mu = \sum_i P_i \tag{3}$$

**To Prove:**

$$\Pr[X > \mu + \delta\mu] \leq e^{(-\delta^2/2+\delta)\mu}$$
$$\Pr[X < \mu - \delta\mu] \leq e^{(-\delta^2/2)\mu}$$

**Proof:** for t > 0,

$$E[e^{tX}] = E[e^{t.\Sigma X_i}]$$
$$E[e^{tX}] = E[\prod_i e^{tX_i}]$$
$$E[e^{tX}] = \prod_i E[e^{tX_i}]$$

Because $X_i$ are independent,

$$E[e^{tX_i}] = (1 - P_i)e^{t.0} + P_i.e^{t.1}$$
$$E[e^{tX_i}] = (1 - P_i)e^{t} + P_i.e^{t}$$
$$E[e^{tX_i}] = 1 + P_i(e^t - 1)$$

We know that, $1 + z \leq e^z$

$$E[e^{tX}] \leq \prod_i e^{P_i(e^t-1)}$$
$$E[e^{tX}] \leq e^{(e^t-1)\sum_i P_i}$$

Applying Equation 1, $E[e^{tX}] \leq e^{(e^t-1)\mu}$
let $A = \mu + \delta\mu$, From Markov's bound

$$\Pr[X > A] = \Pr[e^{tX} > e^{tA}]$$
$$\Pr[X > A] \leq E[e^{tX}]/e^{tA}$$
$$\Pr[X > A] \leq e^{(e^t-1)\mu}/e^{tA}$$

let $t$ be $\ln(1 + \delta)$,

$$\Pr[X > A] \leq e^{(e^{\ln(1+\delta)}-1)\mu}/e^{\ln(1+\delta)A}$$
$$\Pr[X > \mu(1 + \delta)] \leq e^{\mu\delta - A.\ln(1+\delta)}$$
$$\Pr[X > \mu(1 + \delta)] \leq e^{\mu\delta - \mu(1+\delta)\ln(1+\delta)}$$
$$\Pr[X > \mu(1 + \delta)] \leq e^{\mu[\delta - (1+\delta)\ln(1+\delta)]}$$

We know that, $\ln(1+x) < 2x/2 + x => \ln(1+x) < x/(1+x/2)$

$$\Pr[X > \mu(1+\delta)] \leq e^{\mu[\delta-(1+\delta)\delta/(1+\delta/2)]}$$
$$\Pr[X > \mu(1+\delta)] \leq e^{\mu\delta[1-2(1+\delta)/(2+\delta)]}$$
$$\Pr[X > \mu(1+\delta)] \leq e^{\mu\delta[(2+\delta-2(1+\delta))/(2+\delta)]}$$
$$\Pr[X > \mu(1+\delta)] \leq e^{\mu\delta[-\delta/(2+\delta)]}$$
$$\Pr[X > \mu(1+\delta)] \leq e^{(-\delta^2/(2+\delta))\mu}$$

■