1 Hard Core Bits

It is intuitive that it may be possible to concentrate the strength of a one-way function (OWF) into one bit. To develop this idea, we define a function that does this:

Definition 1 \( B : \{0, 1\}^n \rightarrow \{0, 1\} \) is a hard core bit of a OWF \( f : \{0, 1\}^n \rightarrow \{0, 1\}^n \) if:

1. \( B(x) \) is PPT computable
2. \( \forall \) non-uniform PPT \( A \) we have that: \( \Pr_{x,A}[A(f(x)) = B(x)] \leq \frac{1}{2} + \text{neg}(n) \)

Naively, for any OWF, one could believe a particular bit could be used as the hard core bit, but this is not true:

Proof. If \( f \) is a OWF, then we can describe a OWF \( g \) such that: \( \forall i, B_i(x) = x_i \) is not a hard core bit.
Let \( f \) be as above.
Let \( g : \{0, 1\}^{n+\log(n)} \rightarrow \{0, 1\}^{n+\log(n)} \) be defined as
\( g(x, y) = f(x_y) \circ x_y \circ y \) where \( x_y \) is all bits of \( x \) except the \( y \)th bit and similarly \( x_y \) is the \( y \)th bit of \( x \) (here \( \circ \) is used to denote concatenation of the bits)

In Example:
For \( g(000, 0) \rightarrow f(00) \circ 0 \circ 0 \), which reveals the \( x_0 \) (and \( y \)) bit
For \( g(110, 1), x_1 \) is revealed similarly

Construct the \( A_i(f(x_y) \circ x_y \circ y) \) that “breaks” the hard-core bit function such that:

1. if \( y \neq i \) then output a random bit
2. if \( y = i \) then output \( x_y \)

The accuracy of \( A_i \) is \( \frac{n-1}{n} \times \frac{1}{2} + \frac{1}{n} \times 1 = \frac{1}{2} - \frac{1}{2^n} + \frac{1}{n} = \frac{1}{2} + \frac{1}{n} > \frac{1}{2} + \text{neg}(n) \) This algorithm works, since for each \( i \) we have an \( A_i \) that can guess the output of \( B_i \) with greater than \( 1/2 + \text{neg}(n) \) accuracy, but we still cannot guess the entire \( f \) (or \( g \)) non-negligibly
Note that \( g \) is still a OWF, since it keeps the strength of \( f \), but no particular bit (or no particular \( B_i \)) can function as the hard core bit. \( \square \)

A trivial example of a hard core bit that can be constructed from any OWF \( f \) is as follows:
Consider the OWF \( g(b \circ x) = 0 \circ f(x) \) and a hard-core bit function \( B(b \circ x) = b \)
Note that the value \( g(b \circ x) \) does not reveal any information about the first bit \( b \), and hence no information about the value \( B(b \circ x) \) can be ascertained, so, intuitively, the ability for \( A \) to predict the first bit cannot be more than random chance or \( \frac{1}{2} \).
2 One-to-One One-Way Functions and Hard Core Bits

For the remainder of the course we will only be concerned with OWF’s that are one-to-one. We will use the abbreviation OWP for a one-way permutation function (a bijective OWF).

Before we dive into an important characteristic of hard core bits for one-to-one OWF’s, let’s consider a use case for a hard core bit. For example:

Consider two parties trying to perform a coin flip over the phone. How can one party trust the win/loss response from the other party? If one party calls out “heads” and the other responds with “loss”, the second party could be telling a lie. A hard core bit can help with this issue:

Let $f$ be a OWF and $B$ be a hard core bit function for $f$.

Person 1: Sample $x$ randomly from $\{0, 1\}^n$ (or flip $n$ coins) and sends $f(x)$

Person 2: Sends back the choice for the coin - say picking heads and therefore sending back 1

Person 1: Sends back $x, B(x)$. $B(x)$ serves as the “actual” flip of the coin (note that by the definition, it must be difficult to compute from $f(x)$)

If Person 1 lied about the value $x$ and really used $x'$ for the final transmission, then Person 2 would be able to tell since $f(x') \neq f(x)$ from the first transmission.

Let us define the following symbols for the remainder of the lecture to ease discussion about OWF and hard core bits:

Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a OWF

Then let $f' : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$, $f'(x, r) = f(x) \circ r$ have the hard core bit $R(x, r) = \sum_{i=0}^{n} x_i r_i \mod 2$

Note the following general properties: if we are given a one-way function, then we can create a OWF function (as per last lecture). Let $e_i$ be the value/binary string $0\ldots 1\ldots 0$, where the 1 is in the $i$th position.

**Theorem 1** If $\exists$ non-uniform PPT adversary $A$ s.t. $Pr_{x,r,A}[A(f'(x, r)) = B(x, r)] \geq \frac{1}{2} + \epsilon(n)$, where $\epsilon$ is non-neg, then $\exists$ an adversary $R$ that inverts $f$

**Proof.** First, note that $f'$ is a OWF function (as per last lecture). Let $e_i$ be the value/binary string $0\ldots 1\ldots 0$, where the 1 is in the $i$th position.

**Super simple case:**

Assume that $A$ breaks the $B$ with perfect probability/accuracy: $Pr[A(f'(x, r)) = B(x, r)] = 1$

We will now construct an adversary $R(f(x))$ which yields $x$

To invert $f$, $R$:

1. For each $i$, $R$ executes $A(f(x) \circ e^i)$

2. Then $R$ XOR’s (sums, modulo 2) the values from step 1. $A(f(x) \circ e^i) \rightarrow B(x, e^i) = \sum_{j=1}^{n} x_j e^i_j \mod 2 = x_i$

3. $R$ concatenates each $x_i$ and returns the value as $x$
Since $A$ predicts $B(f'(x, r))$ with probability 1, the output of $R$ is produced with probability 1.

To begin with a more complicated case let us consider a set that gives us better probability than that in the theorem - a set that gives $\Pr[E] \geq \frac{1}{2} + \epsilon(n)$ (where $E$ is $A(f'(x, r)) = B(x, r)$ from the theorem statement). Let us define this set as $G$ (standing for Good):

$$\forall x \in G \Pr_{x, r, A}[A(f'(x, r))] = B(x, r) \geq \frac{1}{2} + \frac{\epsilon(n)}{2} \text{ with } \Pr[x \in G] \geq \frac{\epsilon(n)}{2}$$

Assuming $\Pr[x \in G] \leq \frac{\epsilon(n)}{2}$ implies a contradiction, so it is safe to conclude $\Pr[x \in G] \geq \frac{\epsilon(n)}{2}$

**Proof.** Assume that $\Pr[x \in G] \leq \epsilon(n)/2$

$$\frac{1}{2} + \epsilon(n) \leq \Pr_{x, r, A}[E] = \Pr[E(x) \mid x \in G] \times \Pr[x \in G] + \Pr[E(x) \mid x \notin G] \times \Pr[x \notin G]$$

$$< 1 \times \frac{\epsilon(n)}{2} + (\frac{1}{2} + \frac{\epsilon(n)}{2}) \times 1 \leq \frac{\epsilon(n)}{2} + \frac{1}{2} + \frac{\epsilon(n)}{2} = \frac{1}{2} + \epsilon(n) > \frac{1}{2} + \epsilon(n)$$

$$\implies \Pr[x \in G] \geq \frac{\epsilon(n)}{2}$$

Observe that $B(x, r) \oplus B(x, r \oplus e^i) = x_i$

$$= (\sum_j x_j r_j + \sum_j x_j r_j \oplus e^i_j) \mod 2$$

$$= (\sum_j x_j (r_j + x_j r_j) + x_i r_i + x_i (1 - r_i)) \mod 2$$

$$= x_i r_i + x_i - x_i r_i = x_i$$

Let us work with the probability: $\Pr_r [A(f(x), r) \oplus A(f(x), r \oplus e^i) = x_i]$

Note that if both $A$'s guess correctly, we get the right/intended answer. The probability of this happening = both $A$'s are right = 1- either one is wrong

$$\geq 1 - \frac{\text{either one is wrong}}{2(\frac{1}{2} - \epsilon(n))} = 1 - \frac{\text{either one is wrong}}{2\epsilon(n)}$$

**The Simple Case:**

$\Pr[E(x)] \geq \frac{3}{4} + \frac{\epsilon(n)}{2}$

This probability, is bounded by $1 - 2(\frac{1}{4} - \frac{\epsilon(n)}{2}) = \frac{1}{2} + \epsilon(n)$ from the observation before (as one is wrong = $1 - (\frac{3}{4} + \frac{\epsilon(n)}{2}) = \frac{1}{4} - \frac{\epsilon(n)}{2}$)

$R$ then runs the two $A$'s polynomial times and uses majority vote. We use Chebyshev’s inequality to justify the use of majority vote.

### 2.1 Chebyshev’s inequality

Let $x_1, \ldots, x_m$ be independent and identical random variables assuming values 0 or 1. Also, let $\Pr[x_i = 1] = p$.

Then $\Pr[|\sum x_i - pm| > \delta m] < 1/(4\delta^2 m)$

Let $b_1, \ldots, b_T$ be random bits.

Let $X_1$ be 1 when $A(r_1) \rightarrow b_1$.

$X_2$ be 1 when $A(r_2) \rightarrow b_2$.

... 

and let $X_T = 1$ when $A(r_T) \rightarrow b_T$.

Let $T = \frac{2n}{\epsilon(n)^2}$

The problematic case is:
\[
\Pr[\sum_{i=1}^{T} X_i \leq T/2]
= \Pr[\sum_{i=1}^{T} X_i - (\frac{1}{2} + \epsilon(n))T \leq T/2 - (\frac{1}{2} + \epsilon(n)) \times T]
< \Pr[|\sum_{i=1}^{T} X_i - (\frac{1}{2} + \epsilon(n))| > \epsilon(n)\frac{T}{2}]
< \frac{1}{4(\epsilon(n)/2)^2} = \frac{1}{2n} \text{ which is sufficient for the theorem}
\]

In order to show that we can use Chebyshev's inequality, we need to show that the samples we are voting over are pairwise independent:

For any two samples, - x,y - they are pair-wise independent if \( \forall a,b \in \{0,1\} \Pr[x = a \land y = b] = \Pr[x = a] \times \Pr[y = b] \)

Imagine we have \((r_1, B(x, r_1)), \ldots, (r_T, B(x, r_T))\)

Let \( k = \log(T) \), \( S_1, \ldots, S_k \in \{0,1\}^n \) be sampled uniformly, and \( b_1 = B(x, S_1), b_2 = B(x, S_2), \ldots, b_k = B(x, S_k) \ \forall Y \subseteq [k], R \text{ generates } (f(x, \bigoplus_{i \in Y} S_i), \bigoplus_{i \in Y} b_i) \)

(note: \([k]\) is the set \(\{1, \ldots, k\}\))

This proof is finished in the next lecture, with some slight changes.