CS 120: Introduction to Cryptography

Salil Vadhan and Alon Rosen Oct. 19, 2006

Lecture Notes 10:

Hardcore Bits

Recommended Reading.

- Katz-Lindell 6.1.3, 6.3

1 Hardcore Bits

Motivation: If f is a OWF, it is hard to determine x from $f(x)$, but is it also hard to compute a particular bit of x from $f(x)$, say the first bit of x? Random guessing gives a probability of success of $\frac{1}{2}$ but some bits might be even easier to guess. A few examples:

A one-way function can reveal a large part of its input: is there a fraction of the bits of the input which is always "well-hidden"? (i.e. any polynomial-time algorithm cannot have a nonnegligible advantage over random guessing when computing those bits from the output of the function) The answer is no, because we can construct one-way functions such that each bit of x can be obtained from $f(x)$ with high probability. Thus, we instead look for some "bit of information" which is hard to compute.

Definition 1 $b : \{0,1\}^* \to \{0,1\}$ *is a* hardcore bit *(or* hardcore predicate*) for one-way function f if*

- *b is polynomial-time computable.*
- *For every PPT A, there is a negligible function ε such that*

$$\Pr\left[A(f(X)) = b(X)\right] \leq \frac{1}{2} + \varepsilon(n) \qquad \forall n,$$

  where the probability is over $X \xleftarrow{R} \{0,1\}^n$ and the coin tosses of A.

Definition 2 $\{b_{\mathsf{key}} : D_{\mathsf{key}} \to \{0,1\}\}_{\mathsf{key}\in\mathcal{K}}$ *is a* collection of hardcore bits *for the collection of one-way functions $\mathcal{F} = \{f_{\mathsf{key}} : D_{\mathsf{key}} \to R_{\mathsf{key}}\}$ if*

- *Given* $\mathsf{key} \in \mathcal{K}$ *and* $x \in D_{\mathsf{key}}$, $b_{\mathsf{key}}(x)$ *can be computed in polynomial time.*

- *For every PPT A, there is a negligible function ε such that*

$$\Pr\left[A(1^n, K, f_K(X)) = b_K(X)\right] \le \frac{1}{2} + \varepsilon(n) \qquad \forall n,$$

where the probability is taken over $K \stackrel{R}{\leftarrow} G(1^n)$, $X \stackrel{R}{\leftarrow} D_K$, and the coin tosses of A.

2 Examples

RSA functions

- The least significant bit is a hardcore bit for RSA:

$$\mathrm{lsb}_{N,e} : \mathbb{Z}_N^* \mapsto \{0, 1\}$$

  Given $N, e, x^e \bmod N$, we cannot compute $\mathrm{lsb}_{N,e}(x)$ with a nonnegligible advantage over random guessing.

- Define $\mathrm{half}_N(x)$ by $\mathrm{half}_N(x) = 0$ if $0 \le x < N/2$ and 1 otherwise ($\mathrm{half}_N(x)$ is like the most significant bit of x). $\mathrm{half}_N(x)$ is a hardcore bit for RSA.

Rabin's functions

- The least significant bit is a hardcore bit for Rabin's functions:

$$\mathrm{lsb}_N : \mathbb{Z}_N^* \mapsto \{0, 1\}$$

  Given $N, x^2 \bmod N$, we cannot compute $\mathrm{lsb}_N(x)$ with a nonnegligible advantage over random guessing.

- $\mathrm{half}_N(x)$ is a hardcore bit for Rabin's functions.

Modular Exponentiation/Discrete Log $\mathrm{half}_{p-1}(x)$ is a hardcore bit for Modular Exponentiation.

3 Goldreich–Levin hardcore bit

Does every one-way function have a hardcore bit? The following theorem proves that from any arbitrary OWF, we can construct a OWF with a hardcore bit by taking the XOR of a random subset of bits. For $x, r \in \{0, 1\}^n$, define $\langle x, r\rangle = \sum_i x_i r_i \bmod 2 = \oplus_{i|r_i=1} x_i$.

Theorem 3 (Goldreich–Levin hardcore bit) *Let f be any one-way function, and define $f'(x, r) = (f(x), r)$ for $||x|| = ||r||$. Then $\langle x, r\rangle$ is a hardcore bit for f'.*

This theorem is most interesting when f is one-to-one. Note that if f is one-to-one, then so is f'.

Proof ideas:

Reducibility argument: Suppose that there exists a PPT A that predicts $\langle x, r\rangle$ from $(f(x), r)$ with nonnegligible advantage over random guessing. We construct a PPT B that uses A to invert f with nonnegligible probability.

"Easy" case: Assume that $A(f(x), r)$ computes the hardcore bit $\langle x, r\rangle$ with probability 1.

Observation 1: Let $e^{(i)} = (0 \cdots 010 \cdots 0)$ (1 in the i'th position and 0 elsewhere). We observe that $\langle x, e^{(i)}\rangle = x_i$. We define $B(y)$ as follows:

- Let $w_i = A(y, e^{(i)})$ for $1 \leq i \leq n$
- Output $w_1 \cdots w_n$

"Medium" case We assume that $A(f(x), r)$ computes the hardcore bit $\langle x, r\rangle$ with probability $\geq \frac{3}{4} + \varepsilon(n)$, where ε is a nonnegligible function and the probability is taken over the random input x and the coin tosses of A. We have a problem generalizing the argument used in the easy case because A is only guaranteed to succeed on *random* (x, r): we do not know how A behaves if r is not random (such as for $r = e^{(i)}$).

Observation 2: $\langle x, r\rangle \oplus \langle x, r \oplus e^{(i)}\rangle = \langle x, e^{(i)}\rangle = x_i$ because

If r is chosen at random then so is $r \oplus e^{(i)}$.

Attempt #1 to define $B(y)$

- Choose r at random.
- For $1 \leq i \leq n$, compute $w_i = A(y, r) \oplus A(y, r \oplus e^{(i)})$.
- Output $w_1 \cdots w_n$.

$$\Pr_{X,R} A(f(X), R) \neq \langle X, R\rangle \leq \frac{1}{4} - \varepsilon$$

$$\Pr_{X,R} A(f(X), R \oplus e^{(i)}) \neq \langle f(X), R \oplus e^{(i)}\rangle \leq \frac{1}{4} - \varepsilon$$

These two probabilities are not independent so we cannot multiply them together to obtain the probability that $w_i \neq x_i$. Using the Union bound, we get that $\Pr[W_i \neq X_i] \leq \frac{1}{2} - 2\varepsilon$. With this algorithm B, we only expect to recover slightly more than $1/2$ of the bits of x. To avoid this problem, we will repeat the algorithm t times with t random choices of r for each bit of x.

Final algorithm $B(y)$

- Choose $r^{(1)}, r^{(2)}, \cdots, r^{(t)}$ at random $(t = \Theta\left(\frac{n}{\varepsilon^2}\right))$.
- For $1 \leq i \leq n$, define $w_i = \mathrm{maj}\{A\left(y, r^{(j)}\right) \oplus A\left(y, r^{(j)} \oplus e^{(i)}\right) : j = 1, \ldots, t\}$. "maj" means that we take a majority vote over the t trials.
- Output $w_1 \cdots w_n$.

Analysis We cannot immediately apply the Chernoff bound in this case as the probabilities are *not* independent because we are always using the same input y.

A computes $\langle X, R\rangle$ from $(f(X), R)$ (X, R are random variables) with probability of success greater than $\frac{3}{4} + \varepsilon$. This imples that for at least $\varepsilon/2$ fraction of x, $\Pr[A(f(x), R) = \langle x, R\rangle] \geq$

$3/4+\varepsilon/2$ (probability just over R and the coin tosses of A). Call these *good* x. For each good x and each $i \in \{1, \dots, n\}$, $\Pr[A(f(x), R) \oplus A(f(x), R \oplus e_i) \neq x_i] \leq 2 \cdot (1-(3/4+\varepsilon/2)) = 1/2-\varepsilon$.

Thus, the above algorithm inverts f with high probability on $f(x)$ for each good x (for a total success probability of $\approx \varepsilon/2$).

General case (A computes hardcore bit with probability $1/2+\varepsilon$) requires additional ideas.

Theorem 4 (Goldreich-Levin hardcore bit for collections) *Let* $\mathcal{F} = \{f_i : \mathrm{Dom}_i \to \mathrm{Rng}_i\}$ *be any collection of one-way function, and define* $g_{i,r}(x) = f_i(x)$, $b_{i,r}(x) = \langle x, r \rangle$. *Then* $\{b_{i,r} : \mathrm{Dom}_i \to \mathrm{Rng}_i\}$ *is a collection of hardcore bits for the collection of one-way functions* $\{g_{i,r} : \mathrm{Dom}_i \to \mathrm{Rng}_i\}$.