CS355: Topics in cryptography **Fall 2002**

Proof of Yao's XOR Lemma

Yao's XOR lemma is a classic example of amplification of hardness. We give a short proof of the lemma and refer to other surveys on the class web site for a full account.

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation and let $\bar{B} : \{0,1\}^n \to \{0,1\}$ be a (t, ϵ) weak hard-core predicate of f. In other words, there is no t-time algorithm $\mathcal{A}$ for which

$$\Pr[\mathcal{A}(f(x)) = \bar{B}(x) \mid x \leftarrow \{0,1\}^n] > 1 - \epsilon$$

For $m > 0$ define $\bar{B}_m(x_1, \ldots, x_m) = \bar{B}(x_1) \oplus \cdots \oplus \bar{B}(x_m)$. Yao's XOR lemma shows that for sufficiently large m the predicate $\bar{B}_m$ is a hard core predicate of $f_m(x_1, \ldots, x_m) = f(x_1) \| \cdots \| f(x_m)$. Of course, there are more much efficient ways for building hard core bits, as was discussed in class. Neverthrless, the proof Yao's XOR lemma is worth studying for its elegence. We note that Yao's XOR lemma also has applications to an area of learning theory called boosting: given a learning algorithm that "weakly" learns a certain class of functions, the XOR lemma can boost the algorithm to a "strong" learning algorithm for the same class of functions.
We first prove the lemma and then show how it can be used to amplify hardness. In what follows, $B : \{0,1\}^n \to \{0,1\}$ is some hard to compute predicate. For example, $B(x) = \bar{B}(f^{-1}(x))$.

Yao's XOR lemma. Let $B : \{0,1\}^n \to \{0,1\}$ and $B_2(x,y) = B(x) \oplus B(y)$. Suppose $\mathcal{A}_2$ is a probabilistic algorithm satisfying

$$\Pr[\mathcal{A}_2(x,y) = B_2(x,y) \mid x, y \leftarrow \{0,1\}^n] > \frac{1}{2} + \epsilon^2 \tag{1}$$

Then there is a probabilistic algorithm $\mathcal{A}$ whose running time is polynomial in $\mathcal{A}_2$'s running time and satisfies:

$$\Pr[\mathcal{A}(f(y)) = B(y) \mid y \leftarrow \{0,1\}^n] > \frac{1}{2} + \epsilon$$

Proof Sketch. To simplify the notation in the proof we will assume that B is a function from $\{0,1\}^n$ to $\{1,-1\}$ and that $B_2(x,y) = B(x) \cdot B(y)$. From equation 1 one can easily show that

$$E_{x,y}[\mathcal{A}_2(x,y) \cdot B_2(x,y)] > \underbrace{(\frac{1}{2} + \epsilon^2)}_{\Pr[\mathcal{A}(x,y)=B_2(x,y)]} - \underbrace{(\frac{1}{2} - \epsilon^2)}_{\Pr[\mathcal{A}(x,y)\neq B_2(x,y)]} = 2\epsilon^2$$

where $E_{x,y}[\mathcal{A}(x,y) \cdot B_2(x,y)]$ is the expectation of $\mathcal{A}(x,y) \cdot B_2(x,y)$ where x, y are uniformly distributed in $\{0,1\}^n$. Note that $E_{x,y}[\mathcal{A}_2(x,y) \cdot B_2(x,y)]$ is a measure of the correlation between $\mathcal{A}_2$ and B_2.
For $x \in \{0,1\}^n$ define $Q(x) = E_y[B(y) \cdot \mathcal{A}_2(x,y)]$.

Case 1: Suppose there exists $x_0 \in \{0,1\}^n$ such that $|Q(x_0)| > 2\epsilon$. Then $\mathcal{A}(y) \stackrel{def}{=} \mathcal{A}_2(x_0, y)$ satisfies $|E_y[B(y) \cdot \mathcal{A}(y)]| > 2\epsilon$. Then either $E_y[B(y) \cdot \mathcal{A}(y)] > 2\epsilon$ or $E_y[B(y) \cdot \mathcal{A}(y)] < -2\epsilon$. If $E_y[B(y) \cdot \mathcal{A}(y)] > 2\epsilon$ then we have

$$\Pr[\mathcal{A}(y) = B(y) \mid y \leftarrow \{0,1\}^n] > \frac{1}{2} + \epsilon \tag{2}$$

as required. If, $E_y[B(y) \cdot \mathcal{A}(y)] < -2\epsilon$ then $\mathcal{A}'(y) \stackrel{def}{=} -\mathcal{A}(y)$ will satisfy equation (2).

Case 2: Suppose that for all $x \in \{0,1\}^n$ we have that $|Q(x)| \leq 2\epsilon$. Then define algorithm $\mathcal{A}(x)$ as follows:

1. Evaluate $Q(x)$ by sampling a few random $x \in \{0,1\}^n$.
2. Randomly output 1 or -1 so that the expectation of the output is $Q(x)/2\epsilon$. In other words, the algorithm tosses a random biased coin where the probability that the coin falls on its head is $\frac{1}{2}[1 + \frac{Q(x)}{2\epsilon}]$. Indeed, since $|Q(x)| < 2\epsilon$ this value is in $[0,1]$ and is thus a valid probability. If the coin falls on its head we output 1. Otherwise, we output -1.

Then we have:

$$E[\mathcal{A}(x) \cdot B(x)] = E_x[\frac{Q(x)}{2\epsilon} \cdot B(x)] = E_{x,y}[\frac{B(y) \cdot \mathcal{A}_2(x,y)}{2\epsilon} \cdot B(x)] = \frac{1}{2\epsilon} E_{x,y}[B_2(x,y) \cdot \mathcal{A}_2(x,y)] > \epsilon$$

Therefore $\mathcal{A}$ satisfies $\Pr[\mathcal{A}(y) = B(y) \mid y \leftarrow \{0,1\}^n] > \frac{1}{2} + \epsilon$ as required □

To use the lemma, suppose there is no t-time algorithm $\mathcal{A}$ such that

$$\Pr[\mathcal{A}(f(x)) = \bar{B}(x) \mid x \leftarrow \{0,1\}^n] > 1 - \epsilon = \frac{1}{2} + (\frac{1}{2} - \epsilon)$$

The lemma shows that there is no algorithm $\mathcal{A}_2$ such that

$$\Pr[\mathcal{A}_2(f(x), f(y)) = \bar{B}_2(x,y) \mid x, y \leftarrow \{0,1\}^n] > \frac{1}{2} + (\frac{1}{2} - \epsilon)^2 = \frac{3}{4} - 2\epsilon + \epsilon^2$$

and the running time of $\mathcal{A}_2$ is some fixed polynomial function of t. Thus, we get some aplification of hardness (for small ϵ).

Let $m = 2\lceil 1/\epsilon \rceil$. Then iterating the argument $\log_2 m$ times we obtain that there is no algorithm $\mathcal{A}_m$ such that

$$\Pr\left[\mathcal{A}_m\left(f(x_1), \ldots, f(x_m)\right) = \bar{B}_m(x_1, \ldots, x_m)\right] > \frac{1}{2} + (\frac{1}{2} - \epsilon)^m$$

which shows that $\bar{B}_m(x_1, \ldots, x_m)$ is a (t', ϵ) hard core bit of $f(x_1) \| \cdots \| f(x_m)$ as required, where t' is some fixed polynomial function of t.