COM S 6830 – Cryptography September 6, 2011

Lecture 4: Hardness Amplification Theorem

Instructor: Rafael Pass *Scribe: Sujay Jayakar (dsj36)*

1 Preliminaries

Definition 1 *A function $\varepsilon : \mathbb{N} \to \mathbb{R}$ is* **negligible** *if for all c, there exists $n_0 \in \mathbb{N}$ such that for all $n > n_0$, $\varepsilon(n) \leq 1/n^c$. Note the converse: If f is not negligible, there exists a polynomial p such that there exists infinitely many $n \in \mathbb{N}$ such that $f(n) \geq 1/p(n)$.*

Definition 2 *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a* **strong one way function** *if*

- *There exists a PPT algorithm c such that $c(x) = f(x)$ for all $x \in \{0,1\}^*$ (f is easy to compute).*
- *For all nuPPT algorithms A, there exists a negligible ε such that for all $n \in \mathbb{N}$, $\Pr[x \leftarrow \{0,1\}^n : A(1^n, f(x)) \in f^{-1}(f(x))] \leq \varepsilon(n)$ (f is hard to invert).*

Definition 3 *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a* **weak one way function** *if*

- *There exists a PPT algorithm c such that $c(x) = f(x)$ for all $x \in \{0,1\}^*$.*
- *For all nuPPT algorithms A, there exists a polynomial $q(n)$ such that for all $n \in \mathbb{N}$, $\Pr[x \leftarrow \{0,1\}^n : A(1^n, f(x)) \in f^{-1}(f(x))] \leq 1 - \frac{1}{q(n)}$. In other words, the probability an attacker can successfully invert the function is bounded away from one by a non-negligible amount.*

2 The Hardness Amplification Theorem

We will show that the existence of weak one way functions implies the existence of strong way functions via some sort of "amplification" process which turns the easy inversion of the OWF into a much more difficult inversion problem.
One immediate first attempt would be to require that the attacker solve many instances of the weak OWF inversion in parallel. Each weak OWF inversion has a probability of success of $1 - 1/q(n)$, and $n \cdot q(n)$ repetitions would ostensibly have probability of success

$$\left(1 - \frac{1}{q(n)}\right)^{n \cdot q(n)} \approx \left(\frac{1}{e}\right)^n,$$

which is negligible. However, this will not work, as we do not require the attacker to work independently between trials, implying that the probability of failure on the parallel tasks is not necessarily the same. We will have to work a bit harder.

Theorem 1 *Let f be a weak OWF with respect to a polynomial q, and let $m(n) = 2n \cdot q(n)$. Consider $f'(x_1, \ldots, x_{m(n)}) = y_1 \ldots y_{m(n)}$, with $y_i = f(x_i)$. The function f' is a strong OWF.*

Proof. By reduction. Assume for contradiction that f' is not a strong OWF. We will show that this implies that f is not a weak OWF, yielding a contradiction. Assuming f' is not a strong OWF, there exists a nuPPT algorithm A and a polynomial p' such that for infinitely many $n \in \mathbb{N}$, A inverts f' with probability $1/p'(n)$. Or, equivalently, since f' takes inputs of length $n \cdot m(n)$, A inverts f with probability $1/p'(n \cdot m(n))$.
Our goal is to construct an nuPPT algorithm A' from A such that A' can invert f with very high probability ($1 - 1/q(n)$ for some n). One crucial assumption to note is that we are guaranteed that A inverts f' with probability $1/p'(mn)$ *only* when its input is uniformly sampled from $\{0,1\}^{mn}$. We will detail two failed attempts at a construction of A' and the third, successful one.

1. Let $A(x) = A'(x, \ldots, x)$, where we give A' the input string x copied m times. Unfortunately, we have no guarantees on the probability of success in this scenario, as the distribution of inputs is not uniform.

2. Sample $m - 1$ random strings $r_1, \ldots, r_{m-1}$ from $\{0,1\}^n$. Then let

   $$A(x) = A'(x, f(r_1), \ldots, f(r_{m-1})).$$

   This attempt is a little better, but consider an algorithm A' that inverts the first position successfully with probability $1/n$ but always inverts the others. We can do no better than $1/n$ with this construction, which does not suffice.

3. Do the same as in the previous construction, but place x in a random position within the input string.

To implement the third construction, construct an algorithm $A''(y)$ as follows. First sample $i \leftarrow \{1, \ldots, m\}$. Let $y_i = y$, and for $1 \leq j \leq m$, $j \neq i$, let $y_j = f(x_j)$, where $x_j \leftarrow \{0,1\}^n$. Now let $z_1 \ldots z_m = A(y_1 \ldots y_m)$. If $f(z_i) = y$, output z_i and otherwise return $\perp$.

Definition 4 *A string $x \in \{0,1\}^n$ is* **good** *if*

$$\Pr[A''(f(x)) \neq \perp] \geq \frac{1}{2m^2 p'(n)},$$

where $m = m(n)$ is the fixed constant from the theorem statement and p is the polynomial from the negation of the definition of a strong OWF.

Definition 5 *Similarly, a string x is* **bad** *if it is not good. More explicitly,*

$$\Pr[A''(f(x)) \neq \perp] < \frac{1}{2m^2 p'(n)}.$$

Lemma 2 *The number of good $x \in \{0,1\}^n$ is greater than $2^n\left(1-\frac{1}{2q(n)}\right)$, or, equivalently,* $\Pr[X \text{ is bad}] < \frac{1}{2q(n)}$.

Proof. By contradiction. Note that we assumed that A inverts f' with probability $1/p'(n)$. We will show that if $|\text{bad } x| > \frac{2^n}{2q(n)}$, A cannot invert f with such probability. The probability that A succeeds can be split into two cases: There exists some bad x_i, and there are no bad x_i.

$$\begin{aligned}\Pr[A(f'(x_1 \dots x_m)) \text{ succeeds}] &= \Pr[A(f'(x_1 \dots x_m)) \text{ succeeds } \wedge \exists \text{ bad } x_i] \\ &\quad + \Pr[A(f'(x_1 \dots x_m)) \text{ succeeds } \wedge \nexists \text{ bad } x_i]\end{aligned}$$

We will handle the two cases separately. First consider the case where there exists a bad x_i. By the union bound, the probability that A succeeds with a bad x_i is less than or equal to the sum over the probabilities where a *particular* x_i is bad.

$$\begin{aligned}\Pr[A(f'(x_1 \dots x_m)) \text{ succeeds } \wedge \exists \text{ bad } x_i] &\leq \sum_{i=1}^{m} \Pr[A \text{ succeeds } \wedge x_i \text{ bad}] \\ &\leq \sum_{i=1}^{m} \Pr[A \text{ succeeds} | x_i \text{ bad}] \\ &\leq \sum_{i=1}^{m} m \cdot \Pr[A''(f(x_i)) \text{ succeeds} | x_i \text{ bad}]\end{aligned}$$

The second to last inequality follows from expanding the conjunction and dropping the probability that x_i is bad. The final inequality follows from the observation that with probability $1/m$, the random placement in A'' places the input in the same spot. From Definition 5, we have

$$\begin{aligned}\Pr[A(f'(x_1 \dots x_m)) \text{ succeeds } \wedge \exists \text{ bad } x_i] &\leq \sum_{i=1}^{m} \frac{m}{2m^2 p'(n)} \\ &= \frac{1}{2p'(n)}\end{aligned}$$

Now consider the second case, where all the x_i are good.

$$\begin{aligned}\Pr[A \text{ succeeds} | \nexists \text{ bad } x_i] &\leq \Pr[\forall x_i \text{ good}] \\ &\leq \left(1 - \frac{1}{2q(n)}\right)^{2nq(n)} \\ &\approx \left(\frac{1}{e}\right)^n\end{aligned}$$

Therefore,

$$\Pr[A(f'(x_1 \dots x_m)) \text{ succeeds}] \leq \frac{1}{2p'(n)} + \left(\frac{1}{e}\right)^n \leq \frac{1}{p'(n)},$$

implying that A cannot invert f' as assumed, yielding our contradiction. ■

Using our result from Lemma 2, we may now show the existence of an algorithm A' that breaks the weak OWF f. Let $A'(x)$ run A'' on its input $2mn^2p'(n)$ times and return the first result that is not bottom.

$$\begin{aligned}\Pr[x \leftarrow \{0,1\}^n : A'(1^n, f(x)) \text{ fails}] &= \Pr[A'(f(x)) \text{ fails} | X \text{ good}] \cdot \Pr[X \text{ good}] \\ &\quad + \Pr[A'(f(x)) \text{ fails} | X \text{ bad}] \cdot \Pr[X \text{ bad}] \\ &\leq \Pr[A'(f(x)) \text{ fails} | X \text{ good}] + \Pr[X \text{ bad}]\end{aligned}$$

We justify throwing away some of the terms as they are all less than one (being probabilities). By Lemma 2 and Definition 4, we have

$$\begin{aligned}\Pr[x \leftarrow \{0,1\}^n : A'(1^n, f(x)) \text{ fails}] &\leq \left(1 - \frac{1}{2m^2p'(n)}\right)^{2m^2np'(n)} + \frac{1}{2q(n)} \\ &\approx \left(\frac{1}{e}\right)^n + \frac{1}{2q(n)},\end{aligned}$$

which implies that the probability of success is greater than $1 - 1/q(n)$, yielding our contradiction. Therefore, f' must be a strong OWF. ■