CS 600.442 – Modern Cryptography 9 September 2015

Lecture 1: Computational Indistinguishability

Instructor: Abhishek Jain *Scribe: Mason Hemmel*

1 Candidate Weak One-Way function

During the previous class, we established what strong and weak one-way functions were, and we finished class with the intuition that f_{mult} is, in fact, a weak one way function due to the hardness of the factoring problem. During this lecture, we will formalize this intuition and prove that this is indeed true.

Intuition. We know $f_{mult}(x, y) = xy$, but to make xy hard to factor, just any x and y won't do. For instance, if either x or y are even numbers then factoring is trivial. In order to tie the factoring of xy directly to the factoring problem, we will have to make sure that x and y both come from Π_n, the set of primes from 1 to 2^n.

While this is simple to say for a mathematician, this selection appears to give serious issues to a programmer. We can't simply pull primes out of thin air, after all. However, the answer to our problem is closer to pulling from air than we might have guessed!

1.0.1 Chebyshev's Theorem

Chebyshev's theorem states: $For\, x > 1$, $\pi(x) > \frac{x}{2log(x)}$ where $\pi(x)$ is the number of primes $<= x$.[1] A corollary of this theorem is that at least a $\frac{1}{2n}$ fraction of n-bit numbers are prime.

Given this fact along with the fact that primality testing can be done in polynomial time,[2] then we realize that we can simply generate random numbers and check their primality with the guarantee that we will find two primes in polynomial time. With this fact, we may proceed with the formal proof.

1.1 Proof

For the sake of reaching contradiction, we assume there exists A' that inverts $f_{mult}(x, y) = xy$ with probability $P >= 1 - \frac{1}{q(n)}$ for some infinite set of $n \in (N)$ where $q(n)$ is some polynomial. Without loss of generality, we will specify that $q(n) = 8n^2$. Given A', we will now construct an adversary A that breaks our assumption of the hardness of factoring in the following way:

Adversary $A(z)$:

1. Generate $x, y \in 0, 1^n$
2. **if** x and y both prime
3. $\quad z' \leftarrow z$
4. **else**

[1] A proof of this statement can be found on page 43 of the Pass & Shelat lecture notes.
[2] See page 60 of Pass & Shelat.

5. $z' = xy$

6. $w \leftarrow A'(z')$

7. **if** x, y are both prime

8. output w

Since A' and primality testing both are PPT, A is a PPT as well. Further, we note that in A we have replicated the distribution that A' expects (i.e. we must fail when x and y are prime) in order to assure that it works as expected.

But what is the probability that A succeeds? In order to do this, we will analyze when it fails. Specifically, there are two cases in which this occurs:

1. A fails when x and y are not both prime, as it fails to pass our prospective value to our solver. By Chebyshev's theorem, this will happen with $P <= 1 - \frac{1}{4n^2}$
2. A will also fail if A' fails, but we know that this happens with $P <= \frac{1}{8n^2}$

By Union Bound, A will fail with $P <= (1 - \frac{1}{4n^2}) + \frac{1}{8n^2}$, which means that it succeeds with $p <= \frac{1}{8n^2}$. This is a non-negligible probability, which means that A has broken the assumption that factoring is hard, meaning that it cannot exist. Hence, A' cannot exist either, meaning that f_{mult} is indeed a weak one-way function.

2 Intuition on Amplifying Weak One-Way Functions into Strong One-Way Functions

Now that we have established that f_{mult} is a weak one-way function, we now return to the idea that we can make strong one-way functions out of weak one-way functions. In order to begin strengthening these functions, we first have to ask ourselves what makes them weak. As we can see even from the example of f_{mult}, our issue is that while some inputs are challenging, others are trivial. At a high level, we can see that all we need to do is reduce our space of weak inputs.

To begin developing an idea for how this might be done, we will consider the function $g(x_1, x_2) = f(x_1)f(x_2)$. Before we discuss it further, we will attempt to move towards a formalizing of the concept of weakness of a function. Specifically, we will call p the density of strong instances of $f(\cdot)$, with $1 - p$ necessarily being the weak instances of $f(\cdot)$. Also, we note that $p < 1$. Returning to $g(\cdot)$, we can now see that if either instances x_1 or x_2 is hard, then the entire output of $g(\cdot)$ is hard. Continuing this logic, it stands to reason that each time we add an input to $g(\cdot)$, then we halve the remaining "weakness" space. That is to say, by adding a second input, we give ourselves a $\frac{3}{4}$ chance that $g()$ is hard (since a valid inverter for $g(\cdot)$ finds the entire pre-image). By adding a third, we would have a $\frac{7}{8}$ chance, and so on and so forth. Extending this logic out to its conclusion gives a function $g(\cdot)$ with n inputs whose hardness is $1 - (1 - p)^n$.

Formalizing this intuition, however, is much more non-trivial. It will be part of the homework.

3 Introduction to Pseudorandomness

We will begin our discussion of pseudorandomness by introducing various primitives which will lead to a formal introduction.

Definition 1 (Distribution) *We say that X is a distribution over a sample space S if X assigns a probability P_s to each sample $s \in S$ such that* $\sum_{s \in S} P_s = 1$

Now that we know what a distribution is, we might want to ask ourselves if there is a way to formalize the conceptual distance between two or more of them. With an answer, we may be able to find a distribution that, while distinct from the random distribution, resembles it for practical purposes. Moving forward with this intuition, it is clear that for us the similarity must only hold under tests that can be executed in polynomial time relative to the size of the sample the test has. In essence, we want to create a situation in which a given sample of one distribution will be indistinguishable from a sample from another distribution, even if the parent distributions differ. First, we will formalize the concept of an ideal sample size.

Definition 2 (Ensemble of Distributions) *A sequence $\{X_n\}_{n \in \mathbb{N}}$ is called an ensemble if $\forall n \in \mathbb{N}, X_n$ is a probability distribution over $\{0,1\}^{\ell(n)}$ where $\ell(n)$ is some polynomial in n*

Given this formalization, we can then introduce a formal definition of computational indistinguishability.

Definition 3 (Computational Indistinguishability) *Let $\{X_n\}_{n \in \mathbb{N}}$ and $\{Y_n\}_{n \in \mathbb{N}}$ be ensembles such that X_n and Y_n are probability distributions over $\{0,1\}^{\ell(n)}$. We say that these two ensembles are computationally indistinguishable if $\forall$ n.u. PPT distinguishers D, $\exists$ negligible function $\mu(\cdot)$ such that $\forall n \in \mathbb{N}, |Pr[t < -X_n, D(t) = 1] - Pr[t < -Y_n, D(t) = 1| <= \mu(n)$*

Properties of Computational Indistinguishability. The following two properties of computational indistinguishability are extremely useful in proofs.

1. **Robustness:** Polynomial-time transformations over two computationally indistinguishable ensembles preserve computational indistinguishability. That is, if $\{X_n\} \approx \{Y_n\}$, then for any PPT M, $\{M(X_n)\} \approx \{M(Y_n)\}$.
2. **Transitivity:** Given $\{X_n\} \approx \{Y_n\}$ and $\{Y_n\} \approx \{Z_n\}$, it follows that $\{X_n\} \approx \{Z_n\}$. This property is also known as the **Hybrid Lemma**.

Proof. (Robustness Property) By way of contradiction, $\exists D_M$ which distinguishes $M(X_n)$ from $M(Y_n)$ where $M(\cdot)$ is a Turing machine running in polynomial time. Given D_M, we will construct D which distinguishes $\{X_n\}$ from $\{Y_n\}$. The construction of D is quite simple: upon receiving $\{X_n\}$ and $\{Y_n\}$, we will apply $M(\cdot)$ to both. We will pass the resulting values to D_M, which will then distinguish each from the other and pass the result on to $D(\cdot)$. This means that $D(\cdot)$ breaks computational indistinguishability, meaning it cannot exist. Therefore, D_M cannot exist either. Hence, computational indistinguishability is robust. ■

Proof. (Transitivity Property) We will prove a more general statement. Let $X_1, \ldots, X_m$ be probability ensembles such that for every i, $X_i \approx X_{i+1}$. We want to prove that $X_1 \approx X_m$. Suppose that this is not true, i.e., X_1 is distinguishable from X_m. We will then show that there exists $i \in [m-1]$ s.t. X_i is distinguishable from X_{i+1}, which will get us a contradiction.

This proof will proceed using the triangle inequality, which states that $|a+b| <= |a| + |b|$. We may generalize this inequality to $|\sum_{i=1}^{k} x_i| <= \sum_{i=1}^{k} |x_i|$. Suppose that $\mathcal{D}$ distinguishes X_1 and X_m with non-negligible probability $\varepsilon(n)$. Then:

$$|Pr[t \leftarrow X_1 : \mathcal{D}(t) = 1] - Pr[t \leftarrow X_m : \mathcal{D}(t) = 1]| >= \varepsilon(n)$$

Let $g_i = Pr[t \leftarrow X_m : \mathcal{D}(t) = 1|$. Then we can rewrite the above equation as $|g_1 - g_m|$. Given this rewriting, we have

$$\begin{aligned}
\varepsilon(n) &\leq |g_1 - g_m| \\
&= |g_1 - g_2 + g_2 - g_3 + ... + g_{m-1} - g_m| \\
&= |\sum_{i=1}^{m-1} g_i - g_{i+1}| \\
&\leq \sum_{i=1}^{m-1} |g_i - g_{i+1}|
\end{aligned}$$

This is to say that the sum of the $m-1$ values must exceed $\varepsilon(n)$. Hence, there needs to be some i such that

$$|g_i - g_{i+1}| \geq \frac{\varepsilon(n)}{m-1} > \frac{\varepsilon}{m}$$

However, $|g_i - g_{i+1}|$ also happens to be the same probability with which $\mathcal{D}$ distinguishes X_i and X_{i+1}. As a result, $\mathcal{D}$ distinguishes X_i and X_{i+1} with non-negligible probability $\frac{\varepsilon(n)}{m}$, as required ■