CS276: Cryptography September 8, 2015

Pseudorandom Generators

Instructor: Alessandro Chiesa *Scribe: Tobias Boelter*

Context and Summary

In the last lecture we have had a look at the universal one-way function, hardcore predicates for OWFs and the Goldreich-Levin Theorem, which says that $\langle x, r \rangle \bmod 2$ is hardcore for $f(x) \parallel r$, if f is a one-way permutation.

Today we will have a look at the indistinguishably of distributions and pseudorandom generators. Therefore we will introduce the notion of computational indistinguishability that is much more practical than statistical indistinguishability. We will see our first proof by a hybrid argument. Finally we will define the notion of pseudorandom generators and construct a PRG with one bit expansion from one-way permutations.

1 Indistinguishability of Distributions

Definition 1 *Denote by $\mathbf{\Delta}(D)$ the set of all distributions over the domain D.*

Definition 2 *$\mathcal{X}$ is a **family of distributions** over $D(k) \stackrel{def}{:=} \{0,1\}^{n(k)}$, if $\mathcal{X} = \{X_k\}_{k \in \mathbb{N}}$, where each X_k is in $\mathbf{\Delta}(D(k))$. Here $n(k)$ is again a fixed polynomial.*

Definition 3 *A family of distributions $\mathcal{X}$ is **efficiently samplable** if there exists a probabilistic polynomial time machine S such that the random variables $S(1^k)$ and X_k are identically distributed.*

In the following we will discuss, what it means for a families of distributions $\mathcal{X}$ and $\mathcal{Y}$ to be "close".

Definition 4 *Two families of distributions $\mathcal{X}$ and $\mathcal{Y}$ are **statistically indistinguishable**, denoted by $\mathcal{X} \stackrel{s}{\equiv} \mathcal{Y}$, if their statistical distance is negligible in k, where the statistical distance is defined as*

$$\delta(k) \stackrel{def}{:=} \Delta_{TV}(X_k, Y_k) \stackrel{def}{:=} \frac{1}{2} \sum_{z \in D(k)} |\Pr[X_k = z] - \Pr[Y_k = z]|.$$

$\Delta_{TV}(X_k, Y_k)$ is also called the total variation distance and the $\frac{1}{2}$ is for normalization.

Lemma 5 *The total variation distance of two distributions X_k, Y_k can only get smaller if applied to a function f:*

$$\Delta_{TV}(f(X_k), f(Y_k)) \leq \Delta_{TV}(X_k, Y_k)$$

The notion of statistical indistinguishability is very rigid. We want to have a notion of indistinguishability for the case where two distributions are not the same, but nobody in this universe will ever be able to tell the difference. Therefore we change to the computational setting.

Remark 6 *The derandomization of BPP with a pseudorandom generator is direct descendant of the paradigm shift from statistical to computational indistinguishability.*

Definition 7 *Two families of distributions $\mathcal{X}$ and $\mathcal{Y}$ are* ***computationally indistinguishable****, denoted by $\mathcal{X} \stackrel{c}{\equiv} \mathcal{Y}$, if every uniform/non-uniform probabilistic polynomial time distinguisher D has only negligible advantage in distinguishing the distributions:*

$$d(k) :\stackrel{def}{=} |\Pr[D(1^k, X_k) = 1] - \Pr[D(1^k, Y_k) = 1]| \text{ is negligible in } k.$$

The notation $\Pr[D(1^k, X_k) = 1]$ *is an abbreviation for* $\Pr[b = 1 \mid x \leftarrow X_k,\ b \leftarrow D(1^k, x)]$.

Lemma 8 *For all probabilistic polynomial time computable functions f, $\mathcal{X} \stackrel{c}{\equiv} \mathcal{Y} \implies f(\mathcal{X}) \stackrel{c}{\equiv} f(\mathcal{Y})$.*

Proof: Suppose there exists an probabilistic polynomial time distinguisher that can distinguish $f(\mathcal{X})$ from $f(\mathcal{Y})$ with non-negligible probability. Then $D \circ f$ distinguishes $\mathcal{X}$ and $\mathcal{Y}$ with the same probability. □

Remark 9 *If f is non-uniform then $D \circ f$ will also be non-uniform. If we started with non-uniformity in the indistinguishability definition, we also have the possibility to take non-uniform functions here.*

Next, we will prove that if two efficiently samplable families of distributions $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable, then a polynomial number of (independent) samples of $\mathcal{X}$ are computationally indistinguishable from a polynomial number of (independent) samples of $\mathcal{Y}$. We will only look at the proof in the non-uniform case and leave the uniform case as an exercise for the reader.

In the proof we will use a so-called *hybrid argument* that we will see in many proofs from now on.

Theorem 10 *Let $\mathcal{X}$ and $\mathcal{Y}$ be efficiently samplable families that are computationally indistinguishable. Then also $\overline{\mathcal{X}} \stackrel{c}{\equiv} \overline{\mathcal{Y}}$, where $\overline{\mathcal{X}} :\stackrel{def}{=} \{(X_k^{(1)}, \ldots, X_k^{(p(k))})\}_{k\in\mathbb{N}}$ and $\overline{\mathcal{Y}} :\stackrel{def}{=} \{(Y_k^{(1)}, \ldots, Y_k^{(p(k))})\}_{k\in\mathbb{N}}$*

Proof: [by hybrid argument; Goldwasser, Micali] We make the proof by contradiction. Suppose there exists a probabilistic polynomial time distinguisher D such that $d(k)$ is non-negligible. Now for each $i \in \{1, \ldots, p(k)\}$, we define a hybrid random variable $H_k^{(i)}$

$$H_k^{(i)} :\stackrel{def}{=} (X_k^{(1)}, \ldots, X_k^{(i)}, Y_k^{(i+1)}, \ldots, Y_k^{(p(k))}).$$

Notice that $H_k^{(0)} = \overline{\mathcal{Y}}, H_k^{(p(k))} = \overline{\mathcal{X}}$. They are called the extreme hybrids, i.e. those that match the target distributions. The main idea behind the hybrid argument is that if D can distinguish these extreme hybrids, then it can also distinguish neighboring hybrids even though it was not designed to do so.

Let's take $d(k)$ and rewrite it. In the following we will, like often in the future, drop the input 1^k to the distinguisher in the notation for simplicity.

$$\begin{aligned} d(k) &= |\Pr[D(\overline{X_k}) = 1] - \Pr[D(\overline{Y_k}) = 1]| \\ &= |\sum_{i=0}^{p(k)-1} \Pr[D(H_k^i) = 1] - \sum_{i=0}^{p(k)-1} \Pr[D(H_k^{i+1}) = 1]| \\ &\leq \sum_{i=0}^{p(k)-1} |\Pr[D(H_k^i) = 1] - \Pr[D(H_k^{i+1}) = 1]| \end{aligned}$$

That means that there is an index i such that

$$|\Pr[D(H_k^i) = 1] - \Pr[D(H_k^{i+1}) = 1]| \geq \frac{d(k)}{p(k)}.$$

We want to stress again that D was not designed to work on such hybrids in the first place. Nevertheless it does a good job on distinguishing them, at least for a specific index i.

Now we can construct a distinguisher D' for $\mathcal{X}$ and $\mathcal{Y}$ as follows: First sample $z_1, \ldots, z_i$ uniformly random from X_k. Then sample $z_{i+1}, \ldots, z_{p(k)}$ from Y_k and finally return $D(z)$.

We then get

$$|\Pr[D'(X_k) = 1] - \Pr[D'(Y_k) = 1]| = |\Pr[D(H_k^i) = 1] - \Pr[D(H_k^{i+1}) = 1]| \geq \frac{d(k)}{p(k)}$$

in contradiction to the computational indistinguishability of a single sample of $\mathcal{X}$ from a single sample of $\mathcal{Y}$. □

Remark 11 *Let's discuss the uniform vs. non-uniform case. If D is uniform, is D' uniform? That is unclear. D does not know which i to pick, because i is a function of k. We can resolve this issue by just picking i at random. The analysis then works similar and is left to the reader. As a rule of thumb, one should start with the proof for the uniform case and the non-uniform case is often easier.*

To recapitulate what just happened: There are three ingredients in a proof by hybrid argument:

(i) Construct a *polynomial number* of hybrids. The extreme hybrids match the target.

(ii) Under the assumption of a distinguisher for the extreme hybrids, show through averaging that this distinguisher also works for two specific neighboring hybrids.

(iii) From the ability of distinguishing neighbors, construct a distinguisher for single samples, which leads to a contradiction.

We want to stress the danger of writing a faulty proof of theorems similar to this one by induction. Here, induction turns out to be (at least) tricky. Assume in the proof you had an behavior that the size of the distinguisher doubles every time or the advantage lowers every time by half. At the end this would result into a runtime of $2^{p(k)}$ or a advantage of only $frac12^{p(k)}$ respectively. In some sense, argument has to be opened up and not just applied several times "black-box style".

2 Pseudorandomness

With the power of the notion of computational indistinguishability we can now define pseudorandomness. For now denote by U_l the uniform distribution of l-bit strings.

Definition 12 *A family of distributions $\mathcal{X} = \{X_k\}_{k\in\mathbb{N}}$ is called **pseudorandom** if there exists a polynomial $l(n)$ such that $\mathcal{X}$ is computationally indistinguishable from the uniform family $\mathcal{U} = \{U_{l(k)}\}_{k\in\mathbb{N}}$*

It took actually quite long in history to come up with this definition of pseudorandomness. In previous work, Kolmogorov defined a k-bit string to be random if the minimal size of an algorithm that outputs it is k. This notion is very beautiful, but at the same time extremely useless because undecidable. Generally speaking, Cryptography is much more pragmatic.

Next, we will define the notion of pseudorandom generators. Intuitively speaking, a pseudorandom generator is an efficient deterministic algorithm G that stretches a short random seed into a long pseudorandom string.

Definition 13 *A pseudorandom generator (PRG) with output $l(k)$ is a deterministic polynomial time algorithm, such that*

(i) $|G(1^k, s)| = l(|s|)$

(ii) Its output is pseudorandom, i.e. $\{G(1^k, U_k)\}_k \stackrel{c}{\equiv} \{U_{l(k)}\}_k$.

Vital for a PRG to be useful is that $l(k) > k$. If we take $l(k) = k + 1$ this is already non-trivial, but we will now construct such a PRG from a OWP with hardcore predicate.

Theorem 14 *Let f be a one way permutation with hardcore predicate B. Then $G(s) :\stackrel{def}{=} (f(s), B(s))$ is a PRG with one bit expansion.*

Proof: Suppose there exists a probabilistic polynomial time distinguisher such that

$$d(k) = |\Pr[D(f(U_k), B(U_k)) = 1] - \Pr[D(U_{k+1}) = 1]| \text{ is not negligible in } k.$$

We make some rearrangements:

$$\begin{aligned} d(k) &= |\Pr[D(f(U_k), B(U_k)) = 1] - \Pr[D(U_{k+1}) = 1]| \\ &= |\Pr[D(f(U_k), B(U_k)) = 1] - \Pr[D(f(U_k), U_1) = 1]| \\ &= |\Pr[D(f(U_k), B(U_k)) = 1] - \frac{1}{2}\cdot\Pr[D(f(U_k), B(U_k)) = 1] - \frac{1}{2}\cdot\Pr[D(f(U_k), \overline{B(U_k)}) = 1]| \\ &= \frac{1}{2}\cdot|\Pr[D(f(U_k), B(U_k)) = 1] - \Pr[D(f(U_k), \overline{B(U_k)}) = 1]| \end{aligned}$$

Without loss of generality we assume now, that

$$\Pr[D(f(U_k), B(U_k)) = 1] - \Pr[D(f(U_k), \overline{B(U_k)}) = 1] \geq \frac{1}{p(n)}$$

for some polynomial p and infinitely many n. Note that we got rid of the absolute value bars.

Finally, we use D to construct an algorithm A that guesses $B(x)$. Upon input $y = f(x)$ for some x, algorithm A works as follows:

(i) Uniformly choose $\sigma \leftarrow \{0,1\}$.

(ii) Invoke D upon (y, σ).

(iii) If D returns 1, then output σ. Otherwise, output $\bar{\sigma}$.

It yields that the probability of $A's$ success is non negligible:

$$\begin{aligned}
\Pr[A(f(U_n)) = b(U_n)] &= \frac{1}{2} \cdot \Pr[A(f(U_k)) = B(U_k) | \sigma = B(U_k)] + \frac{1}{2} \cdot \Pr[A(f(U_k)) = B(U_k) | \sigma = \overline{B(U_k)}] \\
&= \frac{1}{2} \cdot \Pr[D(f(U_k), B(U_k)) = 1] + \frac{1}{2} \cdot \Pr[D(f(U_k), \overline{B(U_k)}) = 0] \\
&= \frac{1}{2} \cdot \Pr[D(f(U_k), B(U_k)) = 1] + \frac{1}{2}(1 - \Pr[D(f(U_k), \overline{B(U_k)}) = 1]) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \Pr[D(f(U_k), B(U_k)) = 1] - \frac{1}{2} \cdot \Pr[D(f(U_k), \overline{B(U_k)}) = 1]) \\
&\geq \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{p(k)}
\end{aligned}$$

in contradiction to the assumption that b is a hard-core predicate for f. □