CSE 594 : Modern Cryptography 16/02/2017

Lecture 7: Pseudorandomness - I

Instructor: Omkant Pandey *Scribe: Aravind Warrier, Vaishali Chanana*

1 Randomness

Computer needs randomness for many of its operations. For example, randomness is needed for encrypting a session key in an SSL connection or for encrypting a hard drive. The question that arises is how a computer can get randomness. It can use key strokes or mouse movements to generate randomness but it proves out to be uniform as it depends on the entropy of the source.

A natural approach for making any encryption scheme that uses key would be to start off with a short length random key k and expand it to longer *random looking* key k' by using random key generator g such that $k' = g(k)$. One **fundamental question** is can we really expand few random bits into many random bits?

The approach could be good in some cases. For instance, in Goldreich-Levin Theorem, pair-wise independent randomness did the job showing that it could be worked out with smaller randomness. But in cryptography, we need something *as good as truly random*.

2 Pseudorandomness

Let us suppose there are n uniformly random bits: $x = x_1\|...\|x_n$. Pseudorandomness is finding a **deterministic** (polynomial-time) algorithm G such that:

- $G(x)$ outputs a $n+1$ bits: $y = y_1\|...\|y_{n+1}$
- y looks *as good as* a truly random string $r = r_1\|..\|r_{n+1}$

$\{G : \{0,1\}^n \to \{0,1\}^{n+1}\}$ is called a **pseudorandom generator**(PRG) that takes n random bits of x with no additional randomness of its own and outputs $n+1$ random bits of y. This will be discussed in detail later.

What is actually meant by *as good as truly random*? In cryptography, *as good as truly random* means the bits should not follow any pattern. It should also pass all the statistical tests like:

- As many 0s as there are 1s
- Each particular bit is roughly unbiased
- Each sequence of bits occur roughly with same probability

The *main idea* is that no efficient computer can tell $G(x)$ and r apart. In cryptographic language, it means that distributions $\{x \leftarrow \{0,1\}^n : G(x)\}$ and $\{r \leftarrow \{0,1\}^{n+1} : r\}$ are **computationally indistinguishable**.

Before knowing more about pseudorandomness, let us look at some of the underlying definitions that will help us understand the concept better.

2.1 Distribution

X refers to a distribution over sample space S, if it assigns probability p_s to the element $s \in S$ s.t.

$$\sum_s p_s = 1 \tag{1}$$

2.2 Ensembles

Definition 1 *A sequence $\{X_n\}_{n\in\mathbb{N}}$ is called an ensemble if for each $n \in \mathbb{N}$, X_n is a probability distribution over $\{0,1\}^*$.*

Generally, X_n will be a distribution over the sample space$\{0,1\}^{l(n)1}$

3 Computational Indistinguishability

The term *computational indistinguishable* is being used to formalize a way to capture what it means for two distributions X and Y to *look alike* to any efficient test. In short,

$$\text{Efficient test} = \text{Efficient computation} = \text{Non-uniform PPT}$$

Intuition: No non-uniform PPT "distinguisher" algorithm D can tell the two distributions X and Y apart *i.e.* "behavior" of D is same for both of them.

Let us try to figure out the concept through two systems: *scoring system* and *guessing system.*

3.1 Scoring System

Giving non-uniform PPT algorithm D a sample of probability distribution X, calculating score on the D's output:

- $+1$ point if output[2] says "Sample is from X"
- -1 point if output says "Sample is from Y"

For this system, the concept of computationally indistinguishability says that the average score of D on X and Y should roughly be the same. Mathematically, which is:

$$Pr[x \leftarrow X; D(1^n, x) = 1] \approx Pr[y \leftarrow; D(1^n, y) = 1]$$

$$\Rightarrow |Pr[x \leftarrow X; D(1^n, x) = 1] - Pr[y \leftarrow Y; D(1^n, y) = 1]| \leq \mu(n)^3$$

Definition 2 *(Computationally Indistinguishability): Two ensembles of probability distributions $X = \{X_n\}_{n\in N}$ and $Y = \{Y_n\}_{n\in N}$ are said to be computationally indistinguishable if for every non-uniform PPT D, there exists a negligible function $v(.)$ s.t:*

$$\boxed{|Pr[x \leftarrow X_n; D(1^n, x) = 1] - Pr[y \leftarrow Y_n; D(1^n, y) = 1]| \leq v(n)} \tag{2}$$

[1] $l(n)$ is any polynomial in n.
[2] output can be encoded using just one bit: 1 when sample is from X, 0 when sample is from Y
[3] $\mu(n)$ is any negligible function in n

3.2 Guessing system(Prediction advantage)

Giving a non-uniform PPT algorithm D a random sample from either probability distribution X or Y and asking it to guess. D can guess it with probability $\frac{1}{2}$, beyond which D would not be able to.

Definition 3 *(Prediction Advantage): A non-uniform PPT A is said to guess the bit b from the sample t that is picked out of sequence of distribution X_n^b(made with randomness \$) with prediction advantage when $\{\max_A |Pr[b \xleftarrow{\$} 0,1, t \sim X_n^b : A(t) = b] - \frac{1}{2}|\}$ is* ***negligibly close*** *to 0.*

$$|\Pr_b[(A \mid x \leftarrow X_n^b) = b] - \frac{1}{2}| \leq \mu(n) \qquad \forall A \tag{3}$$

3.3 Proof of equivalence(Computationally Indistinguishability $\Leftrightarrow$ Prediction Advantage)

Proof. Starting with value from prediction advantage:

$|Pr[b \leftarrow \{0,1\}; z \leftarrow X^{(b)}; D(1^n, z) = b] - \frac{1}{2}|$

$= |Pr_{x \leftarrow X^1}[D(x) = 1] \cdot Pr[b = 1] + Pr_{x \leftarrow X^0}[D(x) = 0] \cdot Pr[b = 0] - \frac{1}{2}|$

$= \frac{1}{2} \cdot |Pr_{x \leftarrow X^1}[D(x) = 1] + Pr_{x \leftarrow X^0}[D(x) = 0] - 1|$ $\qquad [\because Pr[b = 1] = Pr[b = 0] = \frac{1}{2}]$

$= \frac{1}{2} \cdot |Pr_{x \leftarrow X^1}[D(x) = 1] - \underbrace{(1 - Pr_{x \leftarrow X^0}[D(x) = 0])}_{\text{Prob. of } D \text{ guessing it wrong}}|$

$= \frac{1}{2} \cdot |Pr_{x \leftarrow X^1}[D(x) = 1] - Pr_{x \leftarrow X^0}[D(x) = 1]|$

■

3.4 Formal Statement

Lemma 1 *(Predication Lemma): Let $\{X_n^0\}$ and $\{X_n^1\}$ be ensembles of probability distribution. Let D be a n.u. PPT that $\epsilon(\cdot)$-distinguishes $\{X_n^0\}$ and $\{X_n^1\}$ for infinitely many $n \in \mathbb{N}$. Then, $\exists$ n.u. PPT A s.t.*

$$\boxed{\Pr[b \xleftarrow{\$} \{0,1\}, t \leftarrow X_n^b : A(t) = b] - \frac{1}{2} \geq \frac{\epsilon(n)}{2}} \tag{4}$$

for infinitely many $n \in \mathbb{N}$

3.5 Properties of Computational Indistinguishability

- *Closure:* If we apply an efficient operation X and Y, they remain computationally indistinguishable. i.e., $\forall$ n.u.PPT M

$$\{X_n\} \approx \{Y_n\} \Rightarrow \{M(X_n)\} \approx \{M(Y_n)\}^4 \tag{5}$$

[4] *Notation:* $\{X_n\} \approx \{Y_n\}$ means both the distributions are computationally indistinguishable.

Proof. Lets assume that there exists a n.u. PPT D that that can distinguish between $\{M(X_n)\}$ and $\{M(Y_n)\}$ with non negligible probability μ(n). i.e./

$$|\Pr[t \leftarrow M(X_n) : D(t) = 1] - \Pr[t \leftarrow M(Y_n) : D(t) = 1]| > \mu(n) \tag{6}$$

This implies that:

$$|\Pr[t \leftarrow (X_n) : D(M(t)) = 1] - \Pr[t \leftarrow (Y_n) : D(M(t)) = 1]| > \mu(n) \tag{7}$$

■

- *Transitivity*: Let X^1, X^2,..,X^m, be a sequence of probability distributions. Assume that the machine D distinguishes X^1 and X^2 with probability ϵ. Then

$$\exists\ i\ \in [1, 2, .., m-1] :\ D \textit{ distinguishes } X^i \textit{ and } X^{i+1} \textit{ with probability } \geq\ \frac{\epsilon}{m}. \tag{8}$$

Proof. Assume that there exists a n.u PPT D, which distinguishes X^1 and X^m with probability at least ϵ.

$$|\Pr[t \leftarrow (X_1) : D(t) = 1] - \Pr[t \leftarrow (X_m) : D(t) = 1]| > \mu(n) \tag{9}$$

Let $g_i = \Pr[t \leftarrow X^i : D(t) = 1]$. Since $|g_1 - g_m| > \epsilon$,
$|g_1 - g_2| + |g_2 - g_3| + ... + |g_{m-1} - g_m| > |g_1 - g_2 + g_2 - ... + g_{m-1} + g_m|$

$$\boxed{\therefore \exists\ i,\ s.t., |g_i - g_{i+1}| > \frac{\epsilon}{m}} \tag{10}$$

■

Lemma 2 *(Hybrid Lemma): Let $\{X^0\}$...$\{X^m\}$ where m = poly(n). Suppose there is a n.u PPT D that can distinguish between $\{X^0\}$ and $\{X^m\}$ with a prediction advantage. Then $\exists$ i ∈ [1..,m-1] such that D can distinguish between $\{X^i\}$ and $\{X^{i+1}\}$ at an advantage of at least $\frac{\varepsilon}{n}$.*

Proof. *By the transitivity property of prediction advantages among computational indistinguishable distributions, the prediction advantage between $\{X^0\}$ and $\{X^n\}$ is limited by the sum of all prediction advantages. i.e.*

$$\sum_{i=1}^{n-1} \mu_i = \epsilon \tag{11}$$

$$\boxed{\therefore \exists i, \mu_i \geq \frac{\epsilon}{n}} \tag{12}$$

4 Back to Pseudorandomness

Intuition: A distribution is pseudorandom if it looks like a uniform distribution [5] any n.u PPT.

[5]Notation: Uniform Distribution over $\{0,1\}^{l(n)}$ is denoted by $U_{l(n)}$ or U_l

Definition 4 *An ensemble $\{X_n\}$, where X_n is a distribution over $\{0,1\}^{l(n)}$ is said to be pseudo random if:*

$$\{X_n\} \approx \{U_{l(n)}\} \tag{13}$$

5 Pseudorandom Generators PRG

A Pseudorandom Generator is a computer program which can convert a few random bits into many random bits.

Definition 5 *A deterministic algorithm G is called a pseudorandom generator (PRG) if:*

- *G can be computed in polynomial time.*
- *$|G(x)| > |x|$*
- *$\{x \leftarrow \{0,1\}^{l(n)}: G(x)\} \approx \{U_{l(n)}\}$ where $|l(n)| = |G(0^n)|$*

The stretch of G is defined as $|G(x)|$ - $|x|$

A PRG doesn't have any state associated with it, unlike pseudo random functions, which makes use of a secret key. Another interesting definition of pseudorandomness is that it should pass all the tests that a *truly* random string would pass. One such test is the *Next Bit Test*.

6 Next Bit Test

For a truly random sequence of bits, it is not possible to predict the *next bit* in the sequence with probability better than 1/2 even given all previous bits of the sequence so far. A sequence of bits passes the *next bit test* if no efficient adversary (n.u. PPT) can predict the *next bit* in the sequence with probability better than 1/2 even given all previous bits of the sequence so far.

7 Next-Bit Predictability

Definition 6 *Next-Bit Predictability: An ensemble of distributions $\{X_n\}$ over $\{0,1\}^{l(n)}$ is next-bit unpredictable if, for $\forall i$, $0 \leq i < l(n)$ and a n.u PPT A, $\exists$ negligible function $\upsilon(\cdot)$ s.t:*

$$\boxed{\Pr[t = t_1 ... t_{l(n)} \sim X_n : A(t_1 ... t_i) = t_{i+1}] \leq \frac{1}{2} + \upsilon(n)} \tag{14}$$

8 Next-bit Unpredictability $\iff$ Pseudorandomness

Theorem 3 *If $\{X_n\}$ is next-bit unpredictable then $\{X_n\}$ is pseudorandom.*

Proof. *(Sketch)* Lets assume that it is not. Assume that there is a n.u PPT D which cannot predict what the next bit is, but can identify from which distribution the whole string came from: from a newly constructed one or from uniform. i.e., the Prediction Advantage is noticeable ($> \epsilon$)

Consider the following experiments:
There are two strings $x = x_1x_2..x_{l(n)}$, drawn from the distribution X and $u = u_1u_2..u_{l(n)}$ is drawn from the uniform distribution U.

- $H_0 :=$ D is given $x = x_1x_2..x_{l(n)}$ and $u = u_1u_2..u_{l(n)}$ correctly identifies that it is from x
- $H_1 :=$ D is given $x = x_1x_2..x_{l(n)}$ and $u_1x_2..x_{l(n)}$
  The first bit of x is replaced by the first bit of u. Say, μ_1 is the predicted advantage.
- $H_2 :=$ D is given $u_1x_2..x_{l(n)}$ and $u_1u_2..x_{l(n)}$
  The second bit of x is replaced by the second bit of u. Say, μ_2 is the predicted advantage.
- $H_n :=$ D is given $x = u_1u_2..u_{l(n)-1}x_{l(n)}$ and $x = u_1u_2..u_{l(n)}$
  The last bit of x is replaced by the last bit of u. Say, μ_n is the predicted advantage

By *Hybrid Lemma* $\exists i$ s.t., $\mu_i \geq \frac{\epsilon}{l(n)}$. Using such a n.u. PPT, we can build another n.u PPT which can predict the next bit with non-negligible probability. ■