Notes for Lecture 12 v0.91

These notes are in a draft version. Please give me any comments you may have, because this will help me revise them.

Last Revision 11/30/04

1 Pseudorandom Generators

Definition 1 (Indistinguishability - finite definition) *Two random variables X and Y taking value over $\{0,1\}^n$ are (S,ϵ)-indistinguishable if for every circuit C of size at most S we have*

$$|\mathbf{Pr}[C(X)=1]-\mathbf{Pr}[C(Y)=1]|\leq\epsilon\ .$$

We say that a random variable X is (S,ϵ)-pseudorandom if it is (S,ϵ)-indistinguishable from the uniform distribution.

Definition 2 (Pseudorandom Generator) *A function $G:\{0,1\}^*\rightarrow\{0,1\}^*$ is a pseudorandom generator of stretch $\ell(n)$, where $\ell:\mathbb{N}\rightarrow\mathbb{N}$ and $\ell(n)\geq n+1$, if*

- *G is computable in polynomial time, in the length of the input;*
- *G maps inputs of length n into outputs of length $\ell(n)$;*
- *For every two polynomials $p()$ and $q()$ and for every sufficiently large n, the random variable $G(U_n)$ is $(p(n),1/q(n))$-pseudorandom, where U_n denotes a random variable uniformly distributed in $\{0,1\}^n$.*

Remark 1 *The definition of pseudorandom generators is typically given in the following equivalent form. First, say that a function $\nu:\mathbb{N}\rightarrow\mathbb{R}$ is* negligible *if for every polynomial p and for every sufficiently large n we have $\nu(n)\leq 1/p(n)$. Then G is said to be a pseudorandom generator if it satisfies to the first two properties of the above definition and if, in addition, for every family of polynomial size circuits $\{C_n\}$ there is a negligible function $\nu()$ such that*

$$|\mathbf{Pr}[C_{\ell(n)}(U_{\ell(n)})=1]-\mathbf{Pr}[C_{\ell(n)}(G(U_n))=1]|\leq\nu(n)\ .$$

One can easily verify that this definition implies Definition 2 (note that $\ell(n)$ must be upper bounded by a polynomial). For the other direction, let G be a pseudorandom generator according to Definition 2 and fix a family of polynomial circuits. Define

$$\nu(n):=|\mathbf{Pr}[C_{\ell(n)}(U_{\ell(n)})=1]-\mathbf{Pr}[C_{\ell(n)}(G(U_n))=1]|$$

and note that for every polynomial $q()$ it must be that $\nu(n)\leq 1/q(\ell(n))$ for every sufficiently large n, and so it follows that $\nu()$ must be negligible.

2 One-way Functions and Hard-Core Bits

Definition 3 (One-Way Function – finite definition) *A function $f : \{0,1\}^n \to \{0,1\}^n$ is (S, ϵ)-one-way if for every circuit A of size a most S,*

$$\mathbf{Pr}_{x\in\{0,1\}^n}[f(A(f(x))) = f(x)] \leq \epsilon \ .$$

For the next definition, we adopt the following convention: if $f : \{0,1\}^* \to \{0,1\}^*$ is a function, then we denote by $f_n : \{0,1\}^n \to \{0,1\}^*$ the restriction of f to inputs of length n.

Definition 4 (One-Way Function - asymptotic definition) *A family $f : \{0,1\}^* \to \{0,1\}^*$ is a* one-way function *if:*

- *f is computable in polynomial time (in the length of the input) and*
- *for every two polynomials $p()$ and $q()$ and for every sufficiently large n, f_n is $(p(n), 1/q(n))$-one-way.*

In words, a one-way function is easy to compute but intractable to invert.

Theorem 1 *If pseudorandom generators exist, then one-way functions exist.*

Proof: See Exercises. □

The converse is also true, but it has an extremely difficulty proof.

Theorem 2 ([HILL99]) *If one-way functions exist, then pseudorandom generators exist.*

In these notes we will prove the simpler, but still remarkable, result that if one-way *permutations* exist then pseudorandom generators exist. A one-way permutation is a one-way function f such that, for every n, $f_n : \{0,1\}^n \to \{0,1\}^n$ is a bijection. In general, we call a function $f : \{0,1\}^* \to \{0,1\}^*$ a *permutation* if, for every n, $f_n : \{0,1\}^n \to \{0,1\}^n$ is a bijection.

We begin by introducing the notion of a *hard-core* predicate of a one-way permutation.

Definition 5 (Hard-Core Predicate – finite definition) *A function $B : \{0,1\}^n \to \{0,1\}$ is a (S, ϵ) hard-core predicate for a permutation $f : \{0,1\}^n \to \{0,1\}^n$ if for every circuit A of size at most S we have*

$$\mathbf{Pr}_{x\sim\{0,1\}^n}[A(f(x)) = B(x)] \leq \frac{1}{2} + \epsilon \ .$$

Definition 6 (Hard-Core Predicate – asymptotic definition) *A function $B : \{0,1\}^* \to \{0,1\}$ is a hard-core predicate for a permutation $f : \{0,1\}^* \to \{0,1\}^*$ if*

- *B is computable in polynomial time;*
- *For every two polynomials p and q, and for every sufficiently large n, B_n is $(p(n), 1/q(n))$ hard-core for f_n.*

In words, $B(x)$ is an efficiently computable property of x. Given $f(x)$, however, it is intractable to even guess with probability much better than $1/2$ whether $B(x)$ is zero or one.

For standard conjectured one-way permutations, such as RSA and exponentiation, very simple functions, such as the value of the last bit of the input, are hard-core predicates. The following result shows that every one-way permutation can be modified to have a hard-core predicate.

Theorem 3 (Goldreich-Levin [GL89]) *Let f be a one-way permutation and define f' such that $f'_{2n}(x,r) = f_n(x), r$. Define $B_{2n}(x,r) = x \cdot r$, where $x \cdot r = \sum_i x_i r_i \pmod 2$. Then f' is a one-way permutation and B is a hard-core predicate for f'.*[1]

In words, the theorem says that if f is a one-way permutation, and we pick at random $x \in \{0,1\}^n$ and a subset $S \subseteq \{1,\ldots,n\}$, and we give to an adversary the value $f(x)$ and set S, it is intractable for the adversary to compute $\bigoplus_{i\in S} x_i$, or even to guess such value with probability much better than $1/2$. We defer the proof of the Goldreich-Levin Theorem to a later section.

3 One-way Permutations Imply Pseudorandom Generators

The main result of this section is the following.

Theorem 4 (Blum-Micali-Yao [BM84, Yao82]) *Suppose that one-way permutations exist, and let $\ell(n)$ be a polynomial. Then there are pseudorandom generators of stretch $\ell(n)$.*

We will prove the Theorem in the finite setting, which gives important information about the security of concrete pseudorandom generators based on concrete finite permutations. We begin with the case of stretch $n+1$.

Lemma 5 *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a permutation and $B : \{0,1\}^n \to \{0,1\}$ be a (S,ϵ) hard-core predicate for f. Define*

$$G(x) := f(x), B(x) \ .$$

Then $G(U_n)$ is $(S - O(1), \epsilon)$-pseudorandom.

Proof: We prove that if A is a circuit of size S such that

$$|\mathbf{Pr}_{x\sim\{0,1\}^n, r\sim\{0,1\}}[A(f(x),r)=1] - \mathbf{Pr}_{x\sim\{0,1\}^n}[A(f(x),B(x))=1]| \geq \epsilon \qquad (1)$$

then we can construct a circuit C of size $S + O(1)$ such that

$$\mathbf{Pr}[C(f(x)) = B(x)] \geq \frac{1}{2} + \epsilon$$

[1] An annoying technicality is that the new function is only defined for inputs of even length. One can get around this by saying that when f' gets an input of odd length $2k+1$ it discards the last input bit and then it computes f'_{2k} of the first $2k$ input bits, as defined above.

and this clearly implies that the Lemma is true.

We start by noting that Equation (1) can be rewritten as

$$\mathbf{Pr}_{x\sim\{0,1\}^n, r\sim\{0,1\}}[A'(f(x),r)=1]-\mathbf{Pr}_{x\sim\{0,1\}^n}[A'(f(x),B(x))=1]\geq\epsilon \tag{2}$$

where A' is either A or the complement of A.

Equation (2) means that $A'(f(x),b)$ is more likely to output 1 if $b=B(x)$ than if $b=\neg B(x)$. This suggests the following algorithm.

```
Input: y
// the algorithm receives in input y = f(x) and tries to guess B(x) = B(f^(-1)(y))
begin
   pick random b ∈ {0,1}
   if A'(f(x), b) = 1 then return b
   else return ¬b
end
```

We will prove that, over the choices of x and b, the algorithm, on input $f(x)$ correctly computes $B(x)$ with probability $1/2+\epsilon$. Let us denote by $C_b(y)$ the output of the algorithm given the input y and the random choice b. That is, $C_b(y)=(\neg b)\oplus A'(y,b)$.

$$\begin{aligned}
\mathbf{Pr}[C_b(f(x))=B(x)] &= \mathbf{Pr}[b=B(x)]\cdot\mathbf{Pr}[C_b(f(x))=B(x)|b=B(x)]\\
&\quad+\mathbf{Pr}[b\neq B(x)]\cdot\mathbf{Pr}[C_b(f(x))=B(x)|b\neq B(x)]\\
&= \frac{1}{2}\mathbf{Pr}[A'(f(x),B(x))=1]+\frac{1}{2}\mathbf{Pr}[A'(f(x),\neg B(x))=0]\\
&= \frac{1}{2}+\frac{1}{2}\mathbf{Pr}[A'(f(x),B(x))=1]-\frac{1}{2}\mathbf{Pr}[A'(f(x),\neg B(x))=1]
\end{aligned}$$

Let us now study the last expression. We can think of the probability of the event that $A'(f(x),r)=1$ as the average of the probabilities that $A'(f(x),B(x))=1$ and $A'(f(x),\neg B(x))=1$. Equation (2) tells us that there is a difference of ϵ between the probability of the event $A'(f(x),B(x))=1$ and the event $A'(f(x),r)=1$. Then, it must follow that there is a difference of 2ϵ between the probability of $A'(f(x),B(x))=1$ and of $A'(f(x),\neg B(x))=1$, so that the last expression in the above derivation is at least $1/2+\epsilon$. More formally:

$$\begin{aligned}
&\frac{1}{2}\mathbf{Pr}[A'(f(x),B(x))=1]-\frac{1}{2}\mathbf{Pr}[A'(f(x),\neg B(x))=1]\\
=\ &\mathbf{Pr}[A'(f(x),B(x))=1]-\left(\frac{1}{2}\mathbf{Pr}[A'(f(x),B(x))=1]+\frac{1}{2}\mathbf{Pr}[A'(f(x),\neg B(x))=1]\right)\\
=\ &\mathbf{Pr}[A'(f(x),B(x))=1]-\mathbf{Pr}[A'(f(x),r)=1]
\end{aligned}$$

Combining everything together, we have

$$\mathbf{Pr}[C_b(f(x))=B(x)]=\frac{1}{2}+\mathbf{Pr}[A'(f(x),B(x))=1]-\mathbf{Pr}[A'(f(x),r)=1]\geq\frac{1}{2}+\epsilon$$

Finally, there exists a specific value $b^* \in B$ such that

$$\mathbf{Pr}_x[C_{b^*}(f(x)) = B(x) \geq \frac{1}{2} + \epsilon$$

and we define C to be C_{b^*}. Note that the size of C is $S + O(1)$. $\Box$

Lemma 6 *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a permutation, $B : \{0,1\}^n \to \{0,1\}$ be a (S, ϵ) hard-core predicate for f and suppose that both B and f are computable by circuits of size at most t. Define*

$$G(x) := B(x), B(f(x)), \cdots, B(f^{(k-1)}(x)), f^{(k)}(x) \ .$$

Then $G(U_n)$ is $(S - O(tk), \epsilon/k)$-pseudorandom.

Proof: We again proceed by contradiction. We assume that there is a circuit A of size S such that

$$|\mathbf{Pr}[A(G(x)) = 1] - \mathbf{Pr}[A(r) = 1]| > \epsilon \ , \tag{3}$$

where x is uniform in $\{0,1\}^n$ and r is uniform in $\{0,1\}^{n+k}$, and we show there is a circuit C of size $\leq S + O(tk)$ such that

$$\mathbf{Pr}[C(f(x)) = B(x)] > \frac{1}{2} + \frac{\epsilon}{k}$$

As a first step, we note that there is a circuit A' (which is either equal to A or to its complement) such that Expression (3) can be written as

$$\mathbf{Pr}[A'(G(x)) = 1] - \mathbf{Pr}[A'(r) = 1]| > \epsilon \tag{4}$$

We now do a *hybrid* argument. We define $k+1$ distributions $X_0, \ldots, X_l$. The distribution X_i is defined by computing $g = G(x)$ for a random $x \in \{0,1\}^n$ and picking $r \in \{0,1\}^k$ at random; then the first i bits of r are concatenated with the last $k - i$ bits of g. We have by definition that X_0 is distributed like $G(x)$ and X_k is uniform. (Indeed, since x is uniformly random and f is a permutation, then $f(x)$ is uniformly distributed. By induction, $f^{(i)}(x)$ is uniform for every i.)

We can rewrite Expression (4) as

$$\mathbf{Pr}[A'(X_0) = 1] - \mathbf{Pr}[A'(X_k) = 1] > \epsilon$$

and we note that we can write

$$\begin{aligned} \epsilon &< \mathbf{Pr}[A'(X_0) = 1] - \mathbf{Pr}[A'(X_k) = 1] \\ &= \sum_{j=0}^{k-1} \mathbf{Pr}[A'(X_j) = 1] - \mathbf{Pr}[A'(X_{j+1}) = 1] \end{aligned}$$

and so there exists one j for which

$$\mathbf{Pr}[A'(X_{j-1}) = 1] - \mathbf{Pr}[A'(X_j) = 1] > \frac{\epsilon}{k}$$

which means that A' can distinguish

$$X_{j-1} = b_1, \dots, b_{j-1}, B(f^{(j-1)}(x)), \cdots, B(f^{(k-1)}(x)), f^{(k)}(x)$$

from

$$X_j = b_1, \dots, b_j, B(f^{(j)}(x)), \cdots, B(f^{(k-1)}(x)), f^{(k)}(x)$$

(where b_h are random bits)

Recall that, for every i, the distribution $f^{(i)}(x)$ is uniform in $\{0,1\}^n$. This means that the two distributions above can be equivalently redefined if we substitute $f^{(j)}(x)$ with a uniformly random element y. All this is giving us that C can distinguish

$$b_1, \dots, b_{j-1}, B(f^{(-1)}(y)), B(y), \cdots, B(f^{(k-j-1)}(y)), f^{(k-j)}(y)$$

from

$$b_1, \dots, b_{j-1}, b_j, B(y), \cdots, B(f^{(k-j-1)}(y)), f^{(k-j)}(y)$$

On input y we can compute $f(y), \cdots, f^{(k-j)}(y)$ and also $B(y), \cdots, B(f^{(k-j-1)}(y)$. Consider now the following algorithm.

```
Input: y
// the algorithm receives in input y = f(z) and tries to guess B(z) = B(f^(-1)(y))
begin
  pick random b_1, ..., b_j ∈ {0,1}
  if A'(b_1, ..., b_j, B(y), ..., B(f^(k-j-1)(y)), f^(k-j)(y)) = 1 then return b_j
  else return ¬b_j
end
```

Denote by $C_{b_1,\dots,b_j}(y)$ the output of the algorithm given input y and random choices $b_1, \dots, b_j$. Then, as in the proof of a previous lemma, it is possible to show that

$$\mathbf{Pr}[C_{b_1,\dots,b_j}(f(x)) = B(x)] = \frac{1}{2} + \mathbf{Pr}[A'(X_{j-1}) = 1] - \mathbf{Pr}[A'(X_j) = 1] > \frac{1}{2} + \frac{\epsilon}{k}$$

Finally, we observe that there must exist a fixed choice of $b_1^*, \dots, b_j^*$ such that

$$\mathbf{Pr}[C_{b_1^*,\dots,b_j^*}(f(x)) = B(x)] > \frac{1}{2} + \frac{\epsilon}{k}$$

and we define C to be equal to $C_{b_1^*,\dots,b_j^*}$. $\square$

4 References

The notion of indistinguishability is due to Goldwasser and Micali [GM84], who also introduced the hybrid argument. Blum and Micali [BM84] were the first to give a formal definition of pseudorandom generator, but their definition was not based on indistinguishability. The indistinguishability-based definition is due to Yao [Yao82], who showed the equivalence of his definition to the definition of Blum and Micali. Yao also treated in greater generality the notion of hard-core predicate, that had been used in an ad hoc way by Goldwasser and Micali and by Blum and Micali.

References

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. Preliminary version in *Proc. of FOCS'82*. 3, 7

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 25–32, 1989. 3

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. Preliminary Version in *Proc. of STOC'82*. 7

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 2

[Yao82] A.C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23th IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982. 3, 7

Exercises

1. Let $\{X_n\}_{n\geq 1}$ and $\{Y_n\}_{n\geq 1}$ be ensembles (sets) of random variables, where X_n and Y_n take values over $\{0,1\}^n$. Say that $\{X_n\}$ and $\{Y_n\}$ are *indistinguishable* if for every two polynomials p and q and for every large enough n we have that X_n and Y_n are $(p(n), 1/q(n))$-indistinguishable.

   Prove that if $\{X_n\}$ and $\{Y_n\}$ are computationally indistinguishable, and f is a length-preserving (meaning that the length of the output is always equal to the length of the input) polynomial time computable function, then $\{f_n(X_n)\}$ and $\{f_n(Y_n)\}$ are also computationally indistinguishable.

   [Hint: start by proving that if $f_n()$ is computable by a circuit of size t, and X_n and Y_n are (S,ϵ)-indistinguishable, then $f_n(X_n)$ and $f_n(Y_n)$ are $(S-t,\epsilon)$-indistinguishable.]

2. Prove that there is an ensemble $\{X_n\}$ that is computationally indistinguishable from the ensamble of uniform distributions $\{U_n\}$, even though only $n^{\log n}$ elements of $\{0,1\}^n$ have non-zero probability in X_n.

   [Hint: use the probabilistic method and Chernoff bounds to argue that there exists a random variable X_n that ranges over only $n^{\log n}$ elements of $\{0,1\}^n$ and that is $(n^{\Omega(\log n)}, 1/n^{\Omega(\log n)})$ pseudorandom.]

3. Prove that if pseudorandom generators of stretch $2n$ exist, then one-way functions exist.

   [Hint: prove that the generator itself is a one-way function.]

4. Prove that if a permutation f has a hard-core predicate B, then f is a one-way permutation.

5. Prove that if $\mathbf{P} = \mathbf{NP}$ then there cannot be any pseudorandom generators, even of stretch $n+1$.