Lecture 3

Trapdoor Permutations:

Definition 2.4.5, Domain sampler typically outputs almost uniform distribution on D_i.

Examples of one-way collections:

1. The RSA function:
   a. If we can factor we can invert RSA.
   b. Don't know the converse which is a major open question. And hence as of now, we cannot base "security" of RSA on hardness assumption for factoring.
2. The Rabin function:
   a. Extracting square-root modulo N is equivalent to factoring N.
   b. Example 21
3. The Factoring Permutations based on Blum integers
4. Discrete Logarithms Problem

Natural Candidates for Hard-Core Predicates

Least significant bits of the RSA and Rabin functions. For the DLP collection the predicate
x < P/2 ? is a hard-core.

Trapdoor Permutations Examples: The RSA and Rabin trapdoor.

Pseudorandom Generators:

"Objects are considered computationally equivalent if they cannot be distinguished by any efficient procedure."

Defn 3.2.1 (Probability ensemble)

Defn 3.2.2.(Poly-time Indistinguishability)
Two variants

Lengths of X_n, Y_n and n will be typically polynomially related. Also one can be computed from the other in polynomial time. In such cases 1^n can be omitted from input.

Define "Statistical Difference" (also known as variation distance)

If the ensembles X,Y are statistically close, then are also polynomial-time-indistinguisable

Exercise 6, Converse is not true.

Proposition 3.2.3 : Do Proof, Exercise 10

Comments:

1. One of the distributions above is not polynomial time constructible.
2. If Pseudorandom generators exist then there exists poly-time constructible ensemble that is computationally indistinguishable from the uniform distribution and yet statistically far from it.
3. Rahul's Exercise: Show converse that is

   "A pair of poly-time-constructible ensembles that are both computationally indistinguishable and have a noticeable statistical difference can exist only if Pseudorandom generators exist."
4. Observe that most sequences would actually work. So can also choose sequence with repetition. Hence X_n will be uniform over $2^{n/2}$ distinct elements in the sequence.

Indistinguishability by repeated experiments

Defn 3.2.4
Thm 3.2.6 : Do Proof, Hybrid technique.

- Information theoretic analogue easy to show, Exercise 7

The Hybrid Technique:

1. Extreme hybrids collide with the complex ensembles.
2. Neighboring hybrids are easily related to the basic ensembles.
3. The number of hybrids is small (i.e. polynomial)

Thm 3.2.6. "may" fail if the individual ensembles are "not both" efficiently constructible.

The information theoretic analogue of Thm. 3.2.6 will hold for "any" two ensembles.

Indistinguishability by circuits

Definition 3.2.7

1. Allowing probabilistic circuits in the preceding definition does not increase its power. Exercise 9
2. Indistinguishability by poly-size circuits implies indistinguishability by poly-time algorithms Exercise 10
3. The converse is false Exercise 10.
4. Indistinguishability by poly-size circuits is preserved under repeated experiments even if both ensembles are not efficiently constructible. Exercise 9.

Pseudorandom Ensembles:

Defn 3.2.8

$\{U_n\}_{n \text{ in } N}$ is called the "standard uniform ensemble".
$\{U_{l(n)}\}_{n \text{ in } N}$ where l: N -> N also called the uniform ensemble.
$|X_n|$ is not necessarily n whereas $|U_m| = m$
Pseudorandomness is shorthand for "pseudorandomness with respect to polynomial time".

Exercises : 2, 3, 4, 6, 7, 10, 11

Pseudorandom Generators:

Def 3.3.1, Note that statistical difference between $G(U_n)$ and $U_{l(n)}$ will be large.

Note n+1 <= l(n) <= poly(n)

Pseudorandom generators for expansion $l_1(n)$ exist iff Pseudorandom generators for expansion $l_2(n)$ exist for every n+1 <= $l_1(n)$, $l_2(n)$ <= poly(n)

Increasing the expansion factor: Construction 3.3.2, Theorem 3.3.3 Proof.

Applicability of Pseudorandom Generators:

- The output of a pseudorandom generator can be used as randomness for any efficient algorithm. And the algorithm will not notice the difference.
- In Cryptography often we need lot of "high quality randomness". Pseudorandom generators allow us to produce (resp. exchange and/or share) poly(n) pseudorandom bits at the cost of producing (resp. exchanging and/or sharing) only n random bits !

Pseudorandomness and unpredictability:

Definition 3.3.6(Unpredictability) Theorem 3.3.7 (Pseudorandomness v/s unpredictability)

Comment: The proof implies that if i-bit prefix of H_n^{i+1} pseudorandom and the (i+1)-bit prefix of H_n^{i+1} is unpredictable then the (i+1)-bit prefix of H_n^{i+1} is pseudorandom.

Pseudorandom generators imply one-way function: Proposition 3.3.8.

Exercises: 13, 14, 17, 19, 20

One-way permutations imply Pseudorandom generators

Construction based on single permutation, Preferred presentation. Theorem 3.4.1 two proofs.

Alternative presentation, explain Figure 3.4, Constructions based on Collections of Permutations Explain figure on page 132.

Concrete instantiations, DLF, RSA, squaring modulo N which is a Blum integer.

These procedures requires sampling uniform n-bit primes which using sophisticated methods can be done using n bit of randomness.

Pseudorandom Functions: Motivation, Definition 3.6.1 (Function Ensembles), Definition 3.6.2 (Pseudorandom Function ensembles), Definition 3.6.3 (Efficiently computable Function Ensembles),

Terminology: In the rest of the course, we consider only efficiently computable pseudorandom function ensembles.

Definition 3.6.4 (Alternative formulation of efficiently computable pseudorandom function ensembles)

Construction 3.6.5., explain figure 3.5. State Theorem 3.6.6, Corollary 3.6.7 and Corollary 3.6.8

Proof of Theorem 3.6.6 in next class. Can do Applications: A general methodology.

No Assignment this time :)