Writeup for Thursday, Sept 19

Sadhika Dhanasekar, Dhriti Gampa

September 19, 2024

1 Indistinguishable Rooms

Function f is a psuedorandom function if there's an adversary (A) who cannot distinguish the room they are talking to given the other Room (R) generates random numbers. B (from below in the Theorem) can also emulate R. A can only give an input to both rooms and get an output from one of the function.

$$f_k() \text{ where } k \leftarrow \{0,1\}^n$$

$$R : \{0,1\}^n \to \{0,1\}^n, R \leftarrow RF_n$$

$$\text{PRF} \;\Rightarrow\; \text{CPA-secure encryption}$$

PRG: func g: g(s), $s \leftarrow \{0,1\}^n$ is a random input, U_{2n} is a uniform, 2n-bit random string, and SS_c is a long string generated if g(s) is applied repeatedly.
Therefore, SS_c must be indistinguishable from U_{2n}

2 Theorem: (Goldreich, Goldwaser, Micali, 1984)

$\exists$ PRG $\Rightarrow$ $\exists$ PRF
Construct: Suppose $g : \{0,1\}^n \to \{0,1\}^{2n}$ for all n is PRG.
Binary tree output: $[k \leftarrow 0, 1^n] \to [[g][...]] \to [[g][...]]$ and arrow to $[[g][...]]$ and so on for 2^n strings given n depth. The root node of the binary tree is the initial random string s and each child of the root determined by applying the function g to it's parent node: g(the parent node)

$f_k(x) :=$ value of x^{th} leaf and can be found in O(n) since you follow a single path from the root

Def: $g_0(z) = g(z)[1...n]$
$g_1(z) = g(z)[n+1...2n]$
$f_k(x = x_1x_2...x_n) := g_{x_n} * g_{x_{n-1}} * g_{x_{n-2}} ... * g_x(k)$
Now we must prove this function, f, is part of the PRF family by showing that it is efficiently computable in polynomial time and is secure.
Step 1: f is efficiently computable in polynomial time n because g is computable in polynomial

time n.
Step 2: Assume for contradiction, $\exists$ NUPPT A, poly p, such that for infinitely many $n \in \mathbb{N}$,

$$\Pr[A^{f_k(\cdot)}(1^n) = 1] - \Pr[A^{R(\cdot)}(1^n) = 1] \geq \frac{1}{p(n)}$$

Point: Poly-many (nT) hybrids

$$B(1^n, t), t \in \{0,1\}^n$$

1. $l \leftarrow [n], i \leftarrow [T]$, not 2^l nodes because we can use T
2. First l levels uniform as random function
3. First i query in level l+1 as random function. Use t as (i+1)th query. Remaining queries use GGM.
4. Remaining level > l+1 follow GGM tree
5. Output: $A^{O(\cdot)}(1^n)$, which results in an output of 0 or 1.

Idea: A can only visit poly-many nodes: $T(n) : Time(A)$ so the adversary can choose arbitrary paths depending on where its first query led it. If there are l levels, and l equals 0, then it's just 0 or 1. If l equals n, then we have the original problem.
Hybrid Lemma: $\Rightarrow$ B breaks g w.p. $\geq \frac{1}{p(n)nT(n)}$

$PRG \Rightarrow$ 2n-bit PRG $\Rightarrow$ PRF $\Rightarrow$ CPA-secure
It is not clear how to prove that CPA secure encryption $\rightarrow$ n+1 PRG because not all encryption schemes are random even if they are secure because they could have a fixed string.

3 One Way Functions (OWF)

It is easier to show that CPA-secure encryption implies a One Way Function than to show that CPA-secure encryption implies n+1 PRG. Thus, let's show OWF implies n+1 PRG so that by transitive properly, CPA-secure implies n+1 PRG.

$n+1$ PRG $\rightarrow$ 2n-bit PRG $\rightarrow$ PRF

$\uparrow$ $\downarrow$

OWF $\leftarrow$ CPA-Secure Encryption

If any one of these objects exist, all exist, otherwise none of them exist.
We know about Advanced Encryption Standard (AES): $Enc_k(m)$. Let's say we have one called UVA $Enc_k(x)$. Is AES (similar to RF_n) or UVA greater?