CSA E0 235: Cryptography February 22, 2016

Scribe for Lecture 13

Instructor: Arpita Patra *Submitted by: Nihesh Rathod*

1 Recap of last lecture

The following points show highlights of what we learned during the last lecture.

- We constructed mathematical formulation for **One Way Function (OWF)** also stressed upon the fact that an **OWF** does not exist in realm of an unbounded adversary.
- We tried to construct an **OWF** and in this process also learned that even though an **OWF** should be easy to compute (Condition 1 for a function being an **OWF**), constructing an **OWF** is not an easy task. So, we ended up with learning few example which are not an **OWF**.
- We saw that there doesn't exist an unconditional proof that shows that **OWFs** exist. Proving the existence of **OWF** will also end the decades long debate of N=NP by proving N≠NP.
- We also saw the definition of **One Way Permutation (OWP)** and **Hard-Core Predicates (HCP)**. Again, finding a **HCP** for either **OWF** or **OWP** is not simple.
- We saw that there is no proof which shows that **HCP** exists for any **OWF**. But given an **OWF**, we can construct another **OWF** with its **HCP**. ⇒ **Goldreich-Levin Theorem**
- We concluded our lecture with the non-trivial proof of **Goldreich-Levin Theorem**.

2 Construction of Pseudorandom Generator with minimal expansion from OWP and HCP.

A Pseudorandom Generator is an efficient, deterministic algorithm that maps a shorter 'random string' (seed) to a longer pseudorandom string such that no probablistic polynomial time adversary can differentiate between string given by a PRG and a randomly chosen string from a uniform distribution. PRGs represents an important set of algorithms in Cryptography. Nevertheless, constructing a PRG is not as easy as a walk in a park. Actually, we don't have any proof which unconditionally shows the existence of PRGs. But given an **OWP** F and its **HCP** hc, we can construct a PRG G with expansion factor $n+1$. Let's see how?

Theorem 1 *Let f be an **OWP** with **HCP** hc. Then the algorithm $G(s) = f(s)\|hc(s)$ is a **PRG** with expansion factor $l(n) = n + 1$.*

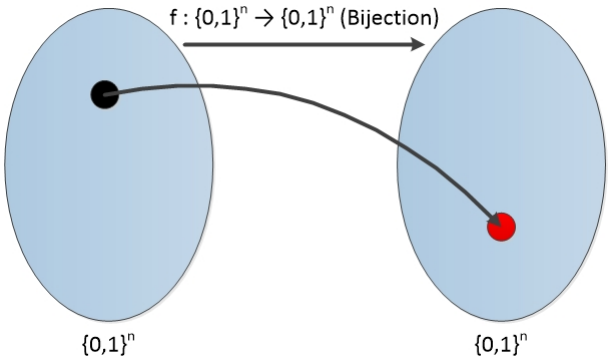

Figure 1: One Way Permutation

We will choose $s \in_R \{0,1\}^n$ and will provide it as input to the **OWP** f. The output of the $f(s)$ will also be uniformly random. Now, given $f(s)$, the value of $hc(s)$ is close to the random.

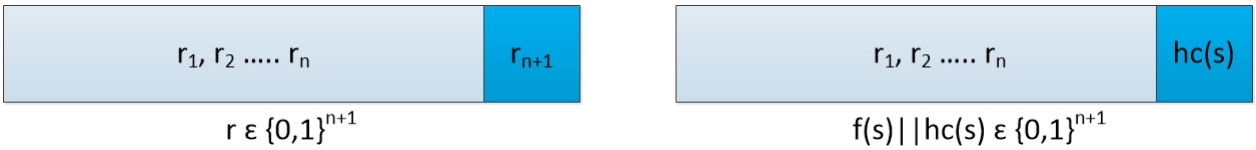

Figure 2: Random string and output of **PRG** G

First n bits in both the strings have same distribution (purely random). Last bit is random in r but it is close to random in output of **PRG** G.

Proof

We wish to prove that $G(s) = f(s)\|hc(s)$ is a **PRG** with expansion factor $l(n) = n+1$.

We are going the prove above theorem by contradiction. The steps of the proof are as follows.

- Assume that $G(s)$ is not a **PRG**. Then there exists a polynomial time adversary D that can distinguish between output of G and a truly random string with probability $\geq neg(n)$.
- We are going to construct an efficient adversary A who can guess the **HCP** $hc(s)$ using $D's$ ability to distinguish between output of G and a truly random string.

- If distinguisher D succeeds then A should be able to guess the **HCP** $hc(s)$ with high probability.
- This will prove that the $hc(s)$ is not a **HCP**, which contradicts with our initial assumptions.

Figure 3: Breaking game for **HCP** $hc(s)$

Figure 3 shows the game for breaking **HCP** $hc(s)$. Adversary A is given value of the function $f(s)$. Now, he has to construct a **HCP** $hc(s)$ for $f(s)$. If he correctly guesses the **HCP** $hc(s)$ then he wins the game otherwise he loses. Now, let's try to calculate distinguisher $D's$ winning probability in terms of **HCP** $hc(s)$.

$$Pr[D(r) = 1] - Pr[D(G(s)) = 1]; r \leftarrow \{0,1\}^{n+1}, s \leftarrow \{0,1\}^n$$
$$= Pr[D(f(s)||r') = 1] - Pr[D(f(s)||hc(s)) = 1]; s \leftarrow \{0,1\}^n, r' \leftarrow \{0,1\}$$
$$= \tfrac{1}{2}Pr[D(f(s)||hc(s)) = 1] + \tfrac{1}{2}Pr[D(f(s)||hc'(s)) = 1] - Pr[D(f(s)||hc(s)) = 1]; s \leftarrow \{0,1\}^n$$
$$= \tfrac{1}{2}Pr[D(f(s)||hc'(s)) = 1] - \tfrac{1}{2}Pr[D(f(s)||hc(s)) = 1]; s \leftarrow \{0,1\}^n$$
$$\geq \tfrac{1}{p(n)}$$

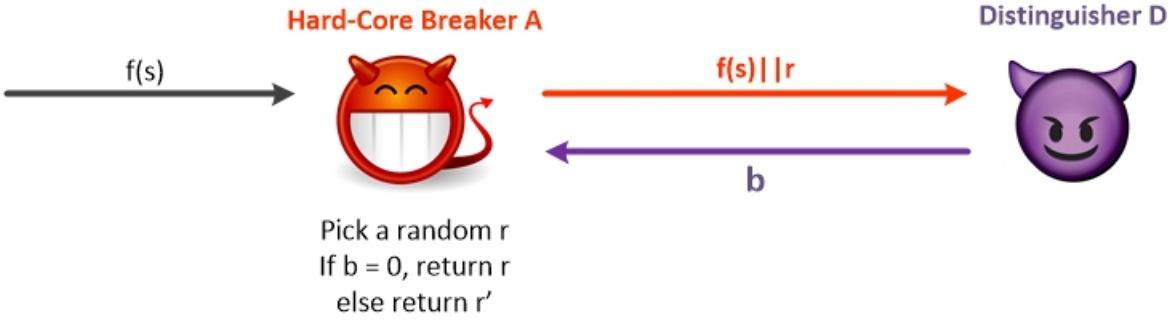

Figure 4: Breaking game for **HCP** $hc(s)$ contd..

Now, Adversary A plays the following trick to convert the **HCP** $hc(s)$ breaking game to the indistinguishability game which distinguisher D excels in breaking. After getting $f(s)$, adversary A picks a random $r \leftarrow \{0,1\}$ and appends it with the $f(s)$ and forwards this $f(s)||r$ to the distinguisher D. Now he waits for the distinguisher D's reply. If D replies

back with $b = 0$ then he returns r as **HCP** $hc(s)$, otherwise he always returns r' as **HCP** $hc(s)$. The same strategy is shown in the above Figure 4. Now let's calculate $A's$ winning probability..

$$
\begin{aligned}
& Pr[A(f(s)) = hc(s)]; s \leftarrow \{0,1\}^n \\
&= Pr[A(f(s)) = hc(s) \wedge r = hc(s)] + Pr[A(f(s)) = hc(s) \wedge r \neq hc(s)]; s \leftarrow \{0,1\}^n \\
&= \tfrac{1}{2}(Pr[A(f(s)) = hc(s) | r = hc(s)] + Pr[A(f(s)) = hc(s) \wedge r | hc(s)]); s \leftarrow \{0,1\}^n \\
&= \tfrac{1}{2}(Pr[D(f(s))||hc(s)) = 0] + Pr[D(f(s))||hc'(s)) = 1]); s \leftarrow \{0,1\}^n \\
&= \tfrac{1}{2} + \tfrac{1}{2}(Pr[D(f(s))||hc'(s)) = 1] - Pr[D(f(s))||hc'(s)) = 1]); s \leftarrow \{0,1\}^n \\
&\geq \tfrac{1}{2} + \tfrac{1}{p(n)}
\end{aligned}
$$

The above probability calculations contradicts our initial assumption that $hc(s)$ is a **HCP**. So, by contradiction we have proved theorem 1. ■

3 PRG with polynomial expansion factor

After constructing a **PRG** with one bit expansion, one obvious question arise. Can we use one bit expansion **PRG** to construct a **PRG** with polynomial expansion factor. Well, that's not an easy task!! Even if we **PRG**, then also proving its existence is a bit tricky. But, still it's worth a try.

Theorem 2 *If there is a **PRG** with expansion factor $l(n) = n + 1$, for any $poly(n)$, then there exists a **PRG** G' with expansion factor $poly(n)$.*

The construction of such a **PRG** and proof of its existence is described in the remaining notes.

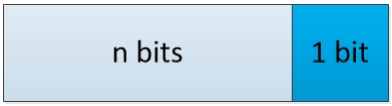

Figure 5: Output of One bit expansion **PRG** G

In the above figure, $G_n(s)$ represents the first n bits of $G(s)$. $G_{n+1}(s)$ represents the first $(n+1)^{th}$ bit of $G(s)$. Using such a **PRG**, we can construct a **PRG** with polynomial expansion factor like the following.

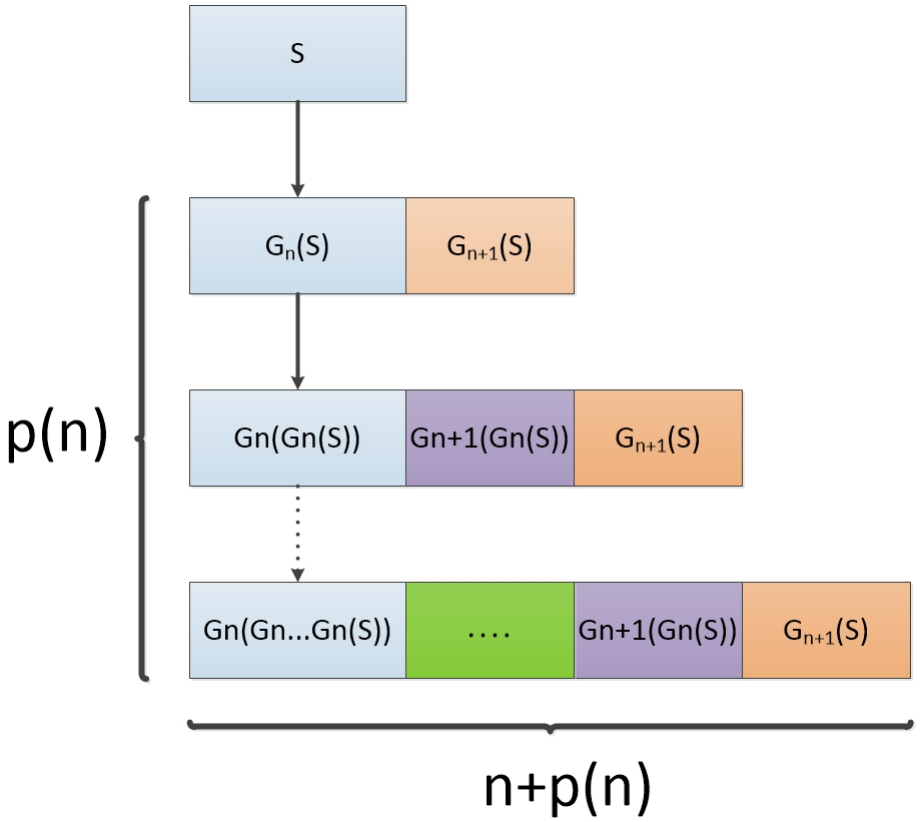

Figure 6: Construction of **PRG** G' with polynomial expansion factor using one bit expansion **PRG** G.

Though the construction is easy to follow and implement, proving that it actually is a **PRG** is not that straight forward. So, we will break the complete proof into smaller problems which are easy to prove. This kind of proving technique is called 'Hybrid arguments'.

Proof For proving the theorem consider the following scenario. Can we come up with a reduction to the distinguisher that distinguishes a random string from a pseudorandom string of length $(n+1)$?

Figure 7: H_0 and $H_{n+p(n)}$ distributions on the leaves for random strings (dark blue color) at 0^{th} level and $(n+p(n))^{th}$ level respectively

The direct answer to above question is 'NO'. There is no direct reduction to this problem. But we can break the above problem into smaller problem which comparatively easy.

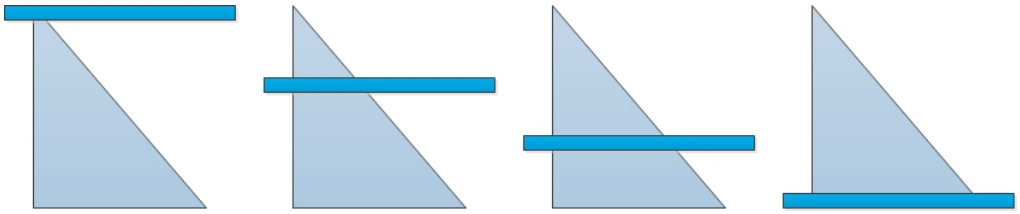

Figure 8: Hybrid steps for problem of figure 7

In the above figure, 1^{st} figure shows the distribution on the leaves H_0 when the 0^{th} level is a random string. Similarly, all the remaining figure shows the distribution on the leaves H_{i-1}, H_i and H_n when the $(i-1)^{th}$, i^{th} and n^{th} level is a random string respectively.

For H_0 and H_1, we need to prove the following..

$$|Pr[D(G'(s)) = 1] - Pr[D(G'(r_1)) = 1]| < negl(n)$$

Similarly, for any successive hybrids, H_{i-1} and H_i, we need to prove the following..

$$|Pr[D(G'(r_{i-1})) = 1] - Pr[D(G'(r_i)) = 1]| < negl(n)$$

When we add probabilities of all the hybrids, probabilities of all the intermediate hybrids will be nullified and we will be left with the following probability.

$$|Pr[D(G'(s)) = 1] - Pr[D(G'(r)) = 1]| < n' * negl(n)$$

which proves that G' is a **PRG**. (**Remark:** It is sufficient to show probability calculations for any $(i-1)^{th}$ and i^{th} hybrids.) For proving the above mentioned probability, we will use the following lemma..

Lemma 3 *If* $G : 0,1^n \rightarrow 0,1^{n+1}$ *is a PRG*

$$|Pr[D(G(s)) = 1] - Pr[A(r) = 1]| \leq negl(n); s \in_R 0,1^n, r \in_R 0,1^{n+1}$$

then the following also holds.

$$|Pr[D(G'(s)) = 1] - Pr[D(r) = 1]| < negl(n); s \in_R 0,1^n, r \in_R 0,1^n$$

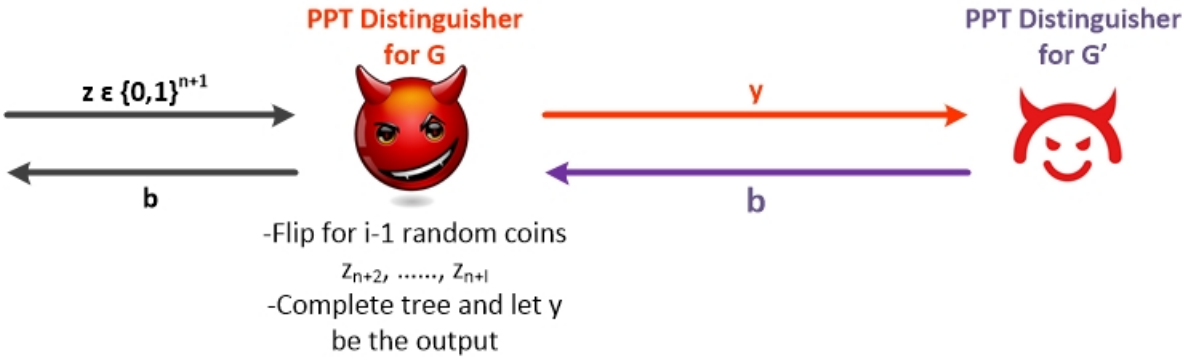

Figure 9: Indistinguishability game for G using distinguisher G'

In the previous lecture, we have seen the detailed proof of the above lemma and we need not to go through it again in this chapter. Assume that there exists a distinguisher G' who can distinguish between H_{i-1} and H_i with probability ¿ $negl(n)$. Let's construct a strategy for PPT distinguisher G who can use above mentioned PPT distinguisher G' for winning the indistinguishability game. One such strategy is shown in the next figure.

When G receives $(n+1)$ length bit string z, he flips $(i-1)$ random coins. Using this and z, he completes the tree and outputs y which is nothing but the distribution on the leave nodes. Now, he waits for the reply from G'. He forwards the reply directly to the challenger. It's easy to follow that G's winning chance is exactly equal to the chances of G'. According to above lemma, we already know that G' has greater than $negl(n)$ probability of winning. When we combine probabilities form all the hybrids, then it clearly shows that G's winning probability is at least $n * negl(n)$ where n is number of hybrid steps. Which is contradicting to our initial assumption.

Hence, finally we have proved that if there is a **PRG** with expansion factor $l(n) = n+1$, for any $poly(n)$, then there exists a **PRG** G' with expansion factor $poly(n)$.

■

References

[1] Ronald Cramer, Ivan Bjerre Damgrd, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing - An Information Theoretic Approach*. Cambridge University Press, 2015.

[2] Arpita Patra. *http://drona.csa.iisc.ernet.in/ arpita/SecureComputation15.html* . Course Materials.