**CSA E0 235: Cryptography** February 20, 2015

Scribe for Lecture 12

*Instructor: Arpita Patra* *Submitted by: Biswajit Nag*

# 1 Prelude

In the last lecture we revisited the notions of pseudo-random functions(PRFs) and pseudo-random generators(PRGs). We saw a construction of a PRF using a length doubling PRG. We were also introduced to the hybrid way of thinking in cryptography, and using this new-gained versatile tool, we proved that the construction we proposed is indeed a PRF.

In today's lecture we will formally define a one-way function. We discuss their current status of existence, we see a few examples of candidate one-way functions. We also define the notion of a hard-core predicate, we see a (partial) proof of their existence. We then revisit the topic of construction of the major theoretical motifs of cryptography, and sign-off by discussing the road-map for constructing a PRG using the ideas of today's lecture.

# 2 One-way Functions

We briefly visited the notion of one-way-ness of functions in Lecture 5. The notion of having functions which are 'easy' to compute but 'difficult' to invert seemed to be a direct way to utilize and limit the probabilistic polynomial time bounded capabilities of real world computing systems.

**Definition 1 (One-way Function)** *A function $f : \{0,1\}^n \mapsto \{0,1\}^m$; $n, m \in \mathbb{N}$ is called a one-way function if the following holds true.*

1. *Computation time for $f(x)$ is polynomially bounded in $n$ $\forall x \in \{0,1\}^n$.*
2. *For all probabilistic polynomial time adversaries $A$ which take an input $y = f(x_0)$, $x_0 \in_R \{0,1\}^n$ and output an $x \in \{0,1\}^n$, there exists corresponding negligible functions neg such that*

$$Pr[x \in f^{-1}(y)] < neg(n)$$

Here we must realize that inversion of a one-way function is easily achievable by using a unbounded powerful adversary.

## 2.1 Some simple non-one-way functions

Negating the conditions for a function being one-way, we observe that a function $f$ is not one-way if the computation

1. there does not exist any polynomial $p$ in $n$ such that $\forall x \in \{0,1\}^n$ the computation time of $f(x)$ is bounded by $p(n)$, or,

2. there exists an adversary $A$ and a polynomial $p$ in $n$ such that

$$\mathsf{Pr}\left[A(f(x), 1^n) \in f^{-1}(f(x))\right]_{x \in_R \{0,1\}^n} \geq \frac{1}{p(n)}$$

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

**Examples**

1. If $f : \{0,1\}^n \mapsto \{0,1\}^m$; $n, m \in \mathbb{N}$ is a function such that there exists an adversary $A$ such that

$$\begin{aligned}\mathsf{Pr}\left[A(f(x), 1^n) \in f^{-1}(f(x))\right]_{x \in_R \{0,1\}^n} &> \frac{1}{n^{10}} \text{ When } n \text{ is odd} \\ &\leq neg(n) \text{ otherwise}\end{aligned}$$

for some negligible function $neg$. Clearly, $f$ is not a one-way function.

2. For $x \in \{0,1\}^n$, the function $f : \{0,1\}^n \mapsto \{0,1\}^{n-1}$; $n \geq 1 \in \mathbb{N}$

$$\begin{aligned}f(x) =& f(x_1 x_2 \cdots x_n) \\ =& x_1 x_2 \cdots x_{n-1}\end{aligned}$$

Here both $x_1, x_2, \cdots x_{n-1}1$ and $x_1, x_2, \cdots x_{n-1}0$ are valid inverses. Hence we have an adversary which produces an inverse with probability 1.

Currently there is no proof for the existence of one-way functions. It's existence will prove the fact that P $\neq$ NP, (as any one-way function will lie in NP but not in P) which itself is considered by many as the most important open problem in theoretical computer science. Here it is to be noted that P $\neq$ NP does not imply the existence of one-way functions, since for a function $f$, it's inversion problem being NP-complete for a particular point in it's range doesn't make it a one-way function, rather it has to be NP-complete for functional values for a proportion of the domain which is at least greater than $1 - neg(n)$ for some negligible function $neg$.

The following are some functions which are believed to be one-way due to a long-standing lack of efficient algorithms for solving them.

- **Subset Sum Problem.**

$$f(x_1, x_2, \cdots, x_n, I) = \sum_{I_j = 1} x_j \bmod 2^n$$

where $I, x_1, x_2, \cdots, x_n \in \{0,1\}^n$.

- **Prime Factorization.**

$$f(p, q) = pq \text{ where } p, q \text{ are primes in } \mathbb{N}.$$

Now we define a slight variant of the one-way function, a one-way permutation. These two are very similar, and almost all the major results about one hold for the other. Our current motivation for defining a one-way permutation is that it will reduce th complexity of the proofs towards the end.

**Definition 2 (One-way Permutation)** *A one-way function $f$ is a one-way permutation if*

- *$f$ is length preserving*
- *$f$ is one-to-one.*

Both the candidate examples for one-way functions we saw before are also candidate one-way permutations (modulo some trivial modifications).

Later in this lecture we use the ideas of one-way functions to construct more (closer to practical) cryptographic functions. One interesting thing to note here is that the notion of one-way functions, despite being the fundamental block supporting the whole of theoretical foundations of cryptography, is surprisingly easy to perceive. This greatly enhances the ease of study of one-way functions. Also, the conditions in the definition of one-way function is rather weak, and we have not made many additional assumptions. This also makes one-way functions a suitable starting point for practical cryptography.

## 3 Hard-core Predicates

When we look at a functional value of a candidate one-way function$f$, we observe that since inversion of $f$ is non-trivial, we will almost certainly not be able to guess a complete and correct inverse. But we might get a lot of information, say, the first half of all the bits, about the input $x$. Stated in a neat way, it is easy to prove that given $f : \{0,1\}^n \mapsto \{0,1\}^m$; $n, m \in \mathbb{N}$ a one-way function, the function $g : \{0,1\}^{2n} \mapsto \{0,1\}^{n+m}$; $n, m \in \mathbb{N}$ and when $x_1, x_2 \in \{0,1\}^m$, $g(x_1||x_2) = x_1||f(x_2)$ is also one way. In general, to keep away an adversary from blindly guessing the inverse just by looking at the functional value, the functional value must conceal at least $O(log(|x|))$ many bits of $x$. Since a one-way function does not reveal the whole input, a hard-core predicate can be thought to be a function which takes the form of some information not revealed in the output. If we model a hard-core predicate as a Boolean function $\{0,1\}^n \mapsto \{0,1\}$ then no probabilistic polynomial timeadversary will be able guess $hc(x)$, the hard-core predicate of $x$ given only the knowledge of $f(x)$, with a probability which is significantly greater than $\frac{1}{2}$.

**Definition 3 (Hard-core Predicate)** *For a given function $f : \{0,1\}^n \mapsto \{0,1\}^m$; $n, m \in \mathbb{N}$, a function $hc : \{0,1\}^n \mapsto \{0,1\}$ is called a hard-core predicate of $f$ if*

1. *$hc(x)$ can be computed within some polynomial time function of the input size.*
2. *For any adversary $A$ which takes the input $f(x), x \in_R \{0,1\}^n$ and outputs a bit , there exists a negligible function neg such that*

$$\mathsf{Pr}\left[A(f(x), 1^n) = hc(x)\right]_{x \in_R \{0,1\}^n} \leq \frac{1}{2} + neg(n)$$

Even if a function is not one-way, it might still have a hard-core predicate. As seen before, $f(x_1, x_2, \cdots x_n) = (x_1, x_2, \cdots x_{n-1})$ (where $\forall i, x_i \in \{0,1\}$) is not a one-way function, but $hc(x_1, x_2, \cdots x_n) = x_n$ is a hard-core predicate for $f$. But since an arbitrary function is

in general of no cryptographic interest, we will consider studying about hard-core predicates only for one-way functions/one-way permutations.

Currently we do not have a proof for or against the fact that every one-way function(or one-way permutation) has a hard-core predicate. A naive argument for the last statement can be that any bit which is not exposed by a one-way function $f$ can be a candidate for it's hard-core predicate. This is not true in general as a one-way function might a reveal all of it's bits (not at a single functional value, obviously) for different inputs from it's domain with finite probability, without exposing too much of information about an inverse at a particular functional value. Also, if $hc$ being a hard-core predicate for a particular one-way function will not make it a hard-core predicatefor all the other one-way functions, as there can be a one-way function precisely revealing $hc$ at all of it's functional values.

Even though we do not have any result about existence of hard-core predicates for one-way functions in general, it is known that for every one-way function, there is a different one-way function with a hard-core predicate. In the next section we state and look at the proving technique for a slightly weaker version of this theorem.

# 4 Existence of one-way permutations with hard-core predicates

**Theorem 4.1 (Golreich-Levin)** *If there exists a one-way permutation* $f : \{0,1\}^n \mapsto \{0,1\}^n, n \in \mathbb{N}$ *then there exists another one-way permutation* $g : \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ *and* $g(x,r) = (f(x),r)$ *where* $x = (x_1, x_2, \cdots, x_n), r = (r_1, r_2, \cdots, r_n) \in \{0,1\}^n$ *with the hard-core predicate* $hc(x,r) = x_1r_1 \oplus x_2r_2 \oplus \cdots \oplus x_nr_n$.

Proving this theorem is equivalent to proving the reduction that given there exists a probabilistic polynomial time adversary $A$ which can break the given candidate hard-core predicate $hc$ with a probability non-negligibly greater than $\frac{1}{2}$ for infinitely many values of $n$, there is another probabilistic polynomial time adversary $A'$ which can invert the one-way permutation$f$ with non-negligible probability for infinitely many $n$.

$$\exists A \text{ s.t. } \mathsf{Pr}\left[A(g(x,r)) = hc(x,r)\right]_{x,r\in_R\{0,1\}^n} \geq \frac{1}{2} + \frac{1}{p_1(n)}$$
$$\Rightarrow \exists A' \text{ s.t. } \mathsf{Pr}\left[A'(f(x)) = x\right]_{x\in_R\{0,1\}^n} \geq \frac{1}{p_2(n)}$$

Where $p_1$ and $p_2$ are polynomials in $n$.

The entire proof of this theorem is beyond the scope of this lecture and was not discussed in it's full completeness. Instead, we prove two similar but weaker results which provide a strong flavour of the reduction used in the original proof.

**Proposition 4.2** *For* $f, g$ *and* $hc$ *defined as in 4.1,* $\forall n \in \mathbb{N}$

$$\exists A \text{ s.t. } \mathsf{Pr}\left[A(g(x,r)) = hc(x,r)\right]_{x,r\in_R\{0,1\}^n} = 1$$
$$\Rightarrow \exists A' \text{ s.t. } \mathsf{Pr}\left[A'(f(x)) = x\right]_{x\in_R\{0,1\}^n} = 1$$

**Proof** Given $A$ for some particular $n \in \mathbb{N}$ we can construct $A'$ for the same $n$ as follows. In the one-way function inverting game if $A'$ is given with $f(x) \in \{0,1\}^n$, then $A'$ queries $A$ with $g(x, e_i) = (f(x), e_i)$ with $i = 1, 2, \cdots n_0$, where $e_i \in \{0,1\}^n$ has 0s in all of it's bits except at the $i^{th}$ one. Since $A$ can give back $hc(x, e_i) = x_i$ with probability 1, in the $n$ queries made $A'$ gets all the bits of $x$ with probability 1. Hence $\mathsf{Pr}\left[A'(f(x)) = x\right]_{x \in_R \{0,1\}^n} = 1$. ■

**Proposition 4.3** *For $f, g$ and $hc$ defined as in 4.1, $\forall n \in \mathbb{N}$, if there is a probabilistic polynomial time adversary $A$ with a polynomial $p(n)$ such that for infinitely many $n \in \mathbb{N}$*

$$\mathsf{Pr}\left[A(g(x,r)) = hc(x,r)\right]_{x,r \in_R \{0,1\}^n} \geq \frac{3}{4} + \frac{1}{p(n)}$$

*then there is another adversary $A'$ such that*

$$\mathsf{Pr}\left[A'(f(x)) = x\right]_{x \in_R \{0,1\}^n} \geq \frac{1}{4p(n)}$$

**Proof**

**Lemma 4.3.1** *Given there exists an adversary $A$ as denoted in 4.3 for some $n$, there exists a subset $S \subset \{0,1\}^n$ such that $|S| \geq \frac{2^n}{2p(n)}$, and $\forall x \in S$*

$$\mathsf{Pr}\left[A(g(x,r)) = hc(x,r)\right]_{r \in_R \{0,1\}^n} \geq \frac{3}{4} + \frac{1}{2p(n)}$$

**Proof** Let $S$ be the set as defined above. Let $U$ denote the set $\{0,1\}^n$. Also, let $S' = U \backslash S$.

$$\begin{aligned}
&\mathsf{Pr}\left[A(g(x,r)) = hc(x,r)\right]_{x,r \in_R \{0,1\}^n} \\
=&\mathsf{Pr}\left[A(g(x,r)) = hc(x,r) | x \in_R S\right]_{r \in_R \{0,1\}^n} \cdot \mathsf{Pr}\left[x \in_R S | x \in_R U\right] \\
&+ \mathsf{Pr}\left[A(g(x,r)) = hc(x,r) | x \in_R S'\right]_{r \in_R \{0,1\}^n} \cdot \mathsf{Pr}\left[x \in_R S' | x \in_R U\right] \qquad (1)
\end{aligned}$$

Since

1. $\mathsf{Pr}\left[A(g(x,r)) = hc(x,r) | x \in_R S\right]_{r \in_R \{0,1\}^n}$ and $\mathsf{Pr}\left[x \in_R S' | x \in_R U\right]$ are at most 1 ;
2. $\mathsf{Pr}\left[A(g(x,r)) = hc(x,r) | x \in_R S'\right]_{r \in_R \{0,1\}^n} \leq \frac{3}{4} + \frac{1}{2p(n)}$ as $x \notin S$;
3. $\mathsf{Pr}\left[A(g(x,r)) = hc(x,r)\right]_{x,r \in_R \{0,1\}^n} \geq \frac{3}{4} + \frac{1}{p(n)}$;
4. $\mathsf{Pr}\left[x \in_R S | x \in_R U\right] = \frac{|S|}{|U|} = \frac{|S|}{2^n}$;

using these in equation 1 we get

$$\frac{3}{4} + \frac{1}{p(n)} \leq \frac{3}{4} + \frac{1}{2p(n)} + \frac{|S|}{2^n} \Rightarrow |S| \geq \frac{2^n}{2p(n)}$$

■

**Lemma 4.3.2** *With $S$ defined in 4.3.1 and $e_i$ defined in 4.2, $\forall x \in S, \forall i = 1, 2, \cdots, n$*

$$\mathsf{Pr}\left[A(g(x,r)) = hc(x,r) \wedge A(g(x, r \oplus e_i)) = hc(x, r \oplus e_i)\right]_{r \in_R \{0,1\}^n} \geq \frac{1}{2} + \frac{1}{p(n)}$$

**Proof**

$$\begin{aligned}
&\mathsf{Pr}\left[A(g(x,r)) = hc(x,r) \wedge A(g(x, r \oplus e_i)) = hc(x, r \oplus e_i)\right]_{r \in_R \{0,1\}^n} \\
=&\mathsf{Pr}\left[A(g(x,r)) = hc(x,r)\right]_{r \in_R \{0,1\}^n} + \mathsf{Pr}\left[A(g(x, r \oplus e_i)) = hc(x, r \oplus e_i)\right]_{r \in_R \{0,1\}^n} \\
&\quad - \mathsf{Pr}\left[A(g(x,r)) = hc(x,r) \vee A(g(x, r \oplus e_i)) = hc(x, r \oplus e_i)\right]_{r \in_R \{0,1\}^n} \\
\geq&\frac{3}{4} + \frac{1}{2p(n)} + \frac{3}{4} + \frac{1}{2p(n)} - 1 \\
=&\frac{1}{2} + \frac{1}{p(n)}
\end{aligned}$$

■

Lemma 4.3.2 tells us that for a random choice of $r \in \{0,1\}^n$ and for a fixed $1 \leq i \leq n$ we will obtain the correct $x_i$ (by taking xor of the two hard-core predicate outputs) with probability significantly greater than $\frac{1}{2}$. Since for a single $r$, the probability of getting the correct $i^{th}$ bit is quite low compared to 1, we repeat this process for multiple independently chosen $r$ values and take the most frequent $x_i$ appearing in these trials as the correct output. We now wish to compute the probability that after running this process $m$ times the true value of $x_i$ is not the most frequent value. Using Chernoff bound and a little bit of probability calculations we get

$$\mathsf{Pr}\left[X_{x_i} < \frac{m}{2}\right] < e^{-\frac{m}{2p^2(n)}}$$

Here $X_{x_i}$ denotes the random variable for the number of outputs of the correct bit, $x_i$, after $m$ such trials.

Since we want this probability to tend to zero for $n \to \infty$, if we choose $e^{-\frac{m}{2p^2(n)}} = \frac{1}{2n}$, we get $m = 2p^2(n)ln(2n)$. (We note that $m$ here is bounded by a polynomial.) Now the probability of failure in getting $x_i$ is at most $\frac{1}{2n}$. Hence after running this algorithm for each $i, 1 \leq i \leq n$, the probability of failure in getting at least one of the bits wrong is bounded below by $\frac{1}{2}$ by the union bound. Hence, for an adversary which guesses each bit $x_i$ using above mentioned procedures for a given $f(x)$ with $x \in_R 0, 1^n$, it's probability of winning is

$$\begin{aligned}
&\mathsf{Pr}\left[A'(f(x)) = x\right]_{x \in_R \{0,1\}^{n_0}} \\
=&\mathsf{Pr}\left[A'(f(x)) = x\right]_{x \in_R S} \cdot \mathsf{Pr}\left[x \in_R S | x \in_R U\right] \\
&\quad + \mathsf{Pr}\left[A'(f(x)) = x\right]_{x \in_R S'} \cdot \mathsf{Pr}\left[x \in_R S' | x \in_R U\right] \\
\geq&\frac{1}{2} \cdot \frac{2^n}{2p(n)} \cdot \frac{1}{2^n} \\
=&\frac{1}{4p(n)}
\end{aligned}$$

■

## 5 Conclusion

In the following road-map of theorems

1. Existence of a one-way permutation imply existence of another one-way permutation with a hard-core predicate.
2. Existence of a one-way permutation with a hard-core predicate imply the existence of a PRG with single bit expansion.
3. Existence of a PRG with single bit expansion imply the existence of PRGs with polynomailly many bit expansions.
4. Existence of a length doubling PRG imply the existence of a PRF.

So far we have proved 1 (partially) in this lecture and 4 (thoroughly) in the previous one . In the following lecture by concatenating a one-way permutation and it's hard-core predicate we construct a PRG of length expansion 1. We expect this to be a proper PRG as for a random seed, since our one-way permutation is a permutation, the one-way permutation value of the seed itself should be a random string with no length expansion. But since the hard-core predicate bit appended to this is almost independent of the one-way permutation functional value (it's independent in the sense that no probabilistic polynomial time adversary will be able to find any causality within these two parts, due to it's hard-core property), the resultant string will look like a random string of length $n+1$, where the seed is of length $n$. Since the step 3 comes almost trivially by repeating this initial PRG generation process, we have now almost completely traversed the road-map for proving the existence of all the major theoretical constructions of cryptography used in this course.

## References


[1] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, 2nd Edition.* CRC Press, 2015.

[2] Arpita Patra. *http://drona.csa.iisc.ernet.in/ arpita/Cryptography16.html* . Course Materials.