CSA E0 235: Cryptography 8^{th} Mar, 2016

Scribe for Lecture 11

Instructor: Arpita Patra *Submitted by: Padma Bhushan Borah*

$\exists$ PRG $\Rightarrow$ $\exists$ PRF:

Introduction

We recall that a length doubling PRG (Pseudo Random Generator) is a deterministic efficient function G from the set of all $n - bit$ strings to that of $2n - bit$ strings such that for random (uniform) distribution on the domain, the distribution defined by G on the output strings is computationally indistinguishable from the uniform distribution on the co-domain. We call a family F of efficient keyed functions, which is a subset of the set $Func$ of all functions from $n - bit$ strings to $n - bit$ strings a PRF (Pseudo Random Function) if for the uniform choice of key of length n, the distribution induced by F on $Func$ is computationally indistinguishable from that of uniform distribution on $Func$.

Since a function in $Func$ can be thought of as a string of length $n.2^n$ bit, we can see that had there been a PRG of expansion factor $n.2^n$, we could have concluded at once the existence of a PRF, and I could have gone home! But (un)fortunately(?) this is not the case :-(So we take shelter in *hybrid arguments* to prove PRG $\Rightarrow$ PRF . The hybrid argument is a proof technique used in crypto in order to show the computational indistinguishability of two distributions, by introducing a series of (at most n-poly many) intermediate hybrid distributions, with the extreme two distributions corresponding to the original distributions. Since poly.negl is again negl, if the consecutive distributions above are computationally indistinguishable, so will be the extreme two distributions. This is what is exploited in hybrid arguments! Now we are ready to give the GGM tree construction: PRF from PRG.

Construction

Suppose $\exists$ a PRG G as described in first para. Let $G_0(x)$ be the first n bits of $G(x)$, and $G_1(x)$ the last n bits of $G(x)$. For each n-bit key k, we construct $F_k \in F := \{F_l \in Func : l \in \{0,1\}^n\}$ as $F_k(x) = F_k(x_1x_2...x_n) := G_{x_n}(G_{x_{n-1}}(...(G_{x_1}(k))...))$. Here k is the seed of the pseudo-random function. We see that the construction can be viewed as a full binary tree of depth n: the root having value k, and for every node with value k' (and so for the root in particular), the left child having value $G_0(k')$, and the right child having value $G_1(k')$. We also note that the the input string can be viewed as the unique path (0 means go left, 1 means go right) traversed from root to leaf, the output being precisely the value of the leaf node. It's left as an exercise for the reader to actually draw the tree:-)

We claim that the above family of functions is a PRF for uniform choice of key k, and devote the remaining *space* and *time* in proving it.

Proof

If NOT, $\exists$ a PPT adversary A, that can distinguish the distribution on $Func$ determined by F for uniform choice of n-bit key k, from the uniform distribution on $Func$. This means A is given access to a function oracle containing f, which is drawn randomly from either of the two distributions on $Func$, and A succeeds after poly(n) query of the function value, in telling apart from which distribution f was drawn with (1/2 + non-negl(n)) probability.

We define $H_i(i = 0, 1, ..., n)$ to be a full binary tree of depth n where the nodes of levels 0 to i are truly random n-bit values, and the levels $i + 1$ to n are constructed by G_0 and G_1. Then H_0 will correspond to F_k for some $k \in \{0,1\}^n$, and H_n will be to a true random function from $Func$. Since A can distinguish H_0 from H_n, by hybrid argument, $\exists j \in \{0, 1, ..., n-1\}$ such that A can distinguish H_j from H_{j+1}; as otherwise A can't distinguish H_0 and H_n too, n being a poly in n. Now, being a PPT algorithm, there are only poly(n) nodes in the j^{th} level of the tree that can be visited by A. Denote all these nodes by $v_1, v_2, ..., v_t$, where $t = poly(n)$. Let $H_{j,i}, (i = 0 \text{ to } t)$, be a full binary tree of depth n where the nodes of levels 0 to j are random, at the $(j+1)^{st}$ level, the children of nodes $v_1, v_2, ..., v_i$ are random, and all other nodes are constructed using G_0 and G_1. Note that $H_{j,0} = H_j$. Also, since PPT A do not visit the nodes in which $H_{j,t}$ and H_{j+1} differ, they are equivalent for A. Hence A can distinguish $H_{j,0}$ from $H_{j,t}$. Therefore, again by our friend 'hybrid argument', $\exists m \in \{0, 1, ..., t-1\}$ such that A can distinguish $H_{j,m}$ from $H_{j,m+1}$. Now we construct an adversary C for G taking input a 2n-bit string S, where S is either a uniform string of length 2n, or a pseudo random string of same length output by G. Given S, consider a full binary tree of depth n, the nodes of levels 0 to j being random n-bit strings, at the $(j+1)^{st}$ level, the children of nodes $v_1, v_2, ..., v_m$ being random, left child of v_{m+1} being first half of S, right child of v_{m+1} being last half of S, and all other nodes being constructed using G_0 and G_1. C gives this tree as input to A. We observe that if S is a uniform string of length 2n, this tree is nothing but from $H_{j,m+1}$, else if S is a pseudo random string output by G, this is from $H_{j,m}$. And since A can distinguish $H_{j,m}$ from $H_{j,m+1}$, so can C a uniform string of length 2n from a pseudo random (output of G is what we are referring to as pseudo random) string of same length- contradicting our assumption that G is a PRG. Thus, the existence of PRG implies the existence of PRF.

Remark:- If there is a PRG with expansion factor $n+1$, so does $\exists$ a PRG with expansion factor Poly(n), for any poly. Therefore WLOG we assumed the expansion factor of our PRG to be 2n in the above proof!

Reference

1. Jonathan Katz, Yehuda Lindell : Introduction to Modern Cryptography, Second Edition, CRC Press.
2. http://www.cs.berkeley.edu/ sanjamg/classes/cs276-fall14/scribe/lec06.pdf
3. Arpita Patra : Lecture Notes, Lecture 11.