Construction of Pseudo-random Functions

Debdeep Mukhopadhyay
IIT Kharagpur

Background

- We have seen how to make Pseudo-random generators from one way functions.
- We shall proceed to make Pseudo-random functions from generators.
- Let G be a PSRG with expansion factor l(n)=2n (i,e G is length doubling)
- Define, G(s)=(G₀(s),G₁(s)), where |s|=|G₀(s)|=|G₁(s)|=n.
• Use G to make keyed function F
  – uses an n bit key
  – takes one bit as input
  – outputs another n bits
• For a key k, define,
  – $F_k(0) = G_0(k)$
  – $F_k(1) = G_1(k)$
• We claim that this is a pseudorandom function! Why?

Simple Reason

• This follows from the fact that G is a pseudorandom generator.
• A random function mapping one bit to n bits is defined by a table of two n-bit values, each of which is random.
• Here we have defined a keyed function, where each n-bit value is pseudorandom (as the key is randomly chosen)
• Thus $F_k$ cannot be distinguished from a random function by a PPT algorithm.
Extend to two bit input

- $F_k(00) = G_0(G_0(k))$
- $F_k(01) = G_1(G_0(k))$
- $F_k(10) = G_0(G_1(k))$
- $F_k(11) = G_1(G_1(k))$

– in order to show that $F_k$ is pseudorandom, thus we have to reason that the four strings, $G_0(G_0(k))$, $G_1(G_0(k))$, $G_0(G_1(k))$, $G_1(G_1(k))$ are pseudorandom.

Hybrid Construction

- $G_0(G_0(k)) \rightarrow G_0(k_0) \rightarrow r_1$
- $G_1(G_0(k)) \rightarrow G_1(k_0) \rightarrow r_2$
- $G_0(G_1(k)) \rightarrow G_0(k_1) \rightarrow r_3$
- $G_1(G_1(k)) \rightarrow G_1(k_1) \rightarrow r_4$

- Here $k_0$, $k_1$, $r_1$, $r_2$, $r_3$ and $r_4$ are randomly chosen $n$ bit strings.
Hybrid Construction

- \( G_0(G_0(k)) \rightarrow G_0(k_0) \rightarrow r_1 \)
- \( G_1(G_0(k)) \rightarrow G_1(k_0) \rightarrow r_2 \)
- \( G_0(G_1(k)) \rightarrow G_0(k_1) \rightarrow r_3 \)
- \( G_1(G_1(k)) \rightarrow G_1(k_1) \rightarrow r_4 \)

If you can distinguish between these strings then you can distinguish either between \( G(k_0) \) and \( (r_1, r_2) \), or \( G(k_1) \) and \( (r_1, r_2) \).

Combining, these facts we have \( F_k \) as pseudorandom.

More generalization

Define: \( F_k : \{0,1\}^n \rightarrow \{0,1\}^n \)

\[ F_k(x_1, x_2, ..., x_n) = G_{x_n}(G_{x_{n-1}}(...G_{x_1}(k))) \]
• The construction can be viewed as a full binary tree of depth n.
• The value at the root is the key k.
• The value of a left child of a node with value k’ is \( G_0(k') \)
• The value of a right child of a node with value k’ is \( G_1(k') \)
• The value of \( F_k(x) \) is thus obtained by traversing the tree according to x
  – if \( x_i = 0 \) traverse left
  – else traverse right
• The entire tree is exponential in n.
  – however to compute the function the entire tree need not be stored. we just need to compute the values on the path and arrive at a leaf.
Theorem

• If $G$ is a pseudorandom generator with expansion factor $l(n) = 2n$, then the above construction is a pseudorandom function.

Proof

• Let $D$ be a PPT algorithm which is given oracle access to a function that is either a random function that maps $n$ bits to $n$ bits, or the function $F_k$ for a randomly chosen $k$. 
Proof

• Consider the distribution of trees, obtained by varying the leaf randomly.
• Each leaf of the binary tree of depth \( n \), is thus a sequence of \( n \) bits.
• use \( H_n^0 \) to denote the distribution.
  – note this is the distribution may be thought of being on the functions \( F_k \).

Proof

• Likewise, define \( H_n^i \) for \( 0 \leq i \leq n \) as follows:
  – values for node \( i \) is chosen at random.
  – values for nodes \( j \geq i+1 \) are chosen as per the function definition. That is see the value of its parent. If the value is \( k' \):
    • value is \( G_0(k') \) if is left child
    • value is \( G_1(k') \) if is right child
  – note that from the point of view of the function, the values of the nodes at levels 0 through \( i-1 \) are irrelevant. This is because they do not decide the value of the leaves.
What is $H_n^n$?

- It is a true random function mapping $n$ bits to $n$ bits.
  - this is because all the leaf values are randomly chosen.
- So, the distinguisher $D$ is able to distinguish between the distribution $H_n^0$ (the actual construction) and $H_n^n$ (the random function)

Construct $D'$ (distinguisher against $G$)

- Assume that $D$ (distinguisher against the PRF $F_k$) makes $t(n)$ queries to the function.
- The output is of length $2n.t(n)$
- Thus $D'$ has $2n.t(n)$ bits of either truly random bits or output generated by $t(n)$ invocations of the function $G(s)$, for a randomly chosen $s$. 
Strategy of D’

- D’ answers queries of D as follows:
  - D asks queries of the form $x_1x_2\ldots x_n$
  - D’ chooses $a_i$ randomly, and goes to node $i$ of the initially empty binary tree.
  - It computes the values of the nodes at level $i+1$ with its sample of length $2n$ as follows:
    - labels the left node with left part of the sample
    - labels the right node with right part of the sample

Observations

- If D’ receives a truly random string of length $2n t(n)$, then it answers D exactly according to $H_{n^{i+1}}$. Why?
- If D’ receives a pseudorandom input, then it answers D exactly according to $H_{n^i}$.
- Thus, if for some $i$, D distinguishes $H_{n^i}$ and $H_{n^{i+1}}$ with a probability of $\epsilon(n)/n$, then with the same probability D’ also distinguishes $t(n)$ invocations of G(s) from a truly random string of length $2n t(n)$ with probability $\epsilon(n)/n$.
- If $\epsilon(n)$ is negligible, we violate the assumption that G is a PRG.
Reading

• How to Construct Random Functions?
  – O. Goldreich, Goldwasser, Micali, JACM 1986

One way functions

• If one way functions then pseudo random generators exist.
• If pseudorandom generators exist, so does pseudorandom functions.
• One way functions are hence necessary.
• Are one way functions sufficient also?
Theorem

• Pseudorandom generators exist only if one-way functions exist or

If there are pseudorandom generators, then there exists one-way functions.

Proof

Let $G$ be a pseudo-random generator with expansion factor of length $2n$. We show $G$ is itself one-way. We shall show that the inability to invert $G$ can be used to distinguish the output of $G$ from random.

Let $A$ be a PPT algorithm, and then define:

$$\varepsilon(n) = \Pr[\text{Invert}_{A,G}(n) = 1]$$

Define $D$ a PPT as follows:

Input: $w \in \{0,1\}^{2n}$

1. Run $A$ on $w$. $x = A(w)$
2. If $w = G(x)$, then return 1, else 0.
Computing the success probability of D.

If \( w \) is random, what is the probability that \( D \) returns 1?

Note that there are at most \( 2^n \) elements in the range of \( G \). If \( w \) falls outside the range, then \( A \) cannot invert and so \( D \) answers 0. Hence, \( \Pr_{w \leftarrow \{0,1\}^n} [D(w)=1] \leq 2^n \)

Conclusion

If \( w = G(s) \) for a uniformly chosen \( s \), then by definition \( A \) computes a correct inverse, with probability exactly \( \varepsilon(n) \). This is the same probability with which \( D \) returns 1.

\[
\therefore \left| \Pr_{w \leftarrow \{0,1\}^n} [D(w) = 1] - \Pr_{x \leftarrow \{0,1\}^n} [D(G(s)) = 1] \right| \geq \varepsilon(n) - 2^n.
\]

Hence, if \( \varepsilon(n) \) is negligible, then \( D \) also have a significant success probability.
Question

• Does secured private key encryption imply the existence of one-way functions?
  – not straightforward
  – there may be construction techniques which do not depend on the above primitives.

• We show that it really does, assuming the weakest form of security notions of the encryption scheme.

Theorem

• If there exists a private key encryption that has indistinguishable encryptions in the presence of an eavesdropper, the one-way functions exist!
  – note for a perfect cipher, where the key length is same or more than the message length, such an assumption need not hold.
  – so we are considering practical ciphers, where the key length is less than the message length.
Proof

Define $\Pi=(\text{Gen},\text{Enc},\text{Dec})$ be a private key encryption scheme that has indistinguishable encryptions in the presence of an adversary. Define $f: f(k,m,r)=(\text{Enc}_k(m,r),m)$ Here $k$, $m$ and $r$ are respectively of $n$, $2n$ and $l(n)$ bits. That is the encryption uses at most $l(n)$ bits of randomness. We claim that this function is one-way.

Proof

Consider a PPT algorithm $A$, which inverts the function, $f$ with a probability of $\varepsilon(n)$.
$\therefore \varepsilon(n)=\Pr[\text{Invert}_{A,f}(n)=1]$. Assume $\varepsilon(n)$ is non-negligible.

Now define a PPT algorithm $A'$, which runs an experiment $\text{Priv}_{A',f}^{\text{CPA}}(n)$.
Now define a PPT algorithm $A'$, which runs an experiment $\text{Priv}_{\text{CPA}}^{\text{A',A}}(n)$.

1. $A'$ chooses random $m_0, m_1 \leftarrow \{0,1\}^n$ and output the two messages. It receives a challenge $c$, which is the encryption of $m_b$, where $b$ is randomly chosen.

2. $A'$ has to say whether $b=0$ or 1. $A'$ runs $A(c,m_b)$ to obtain $(k',m',r')$. If $f(k',m',r')=(c,m_b)$, then $A'$ outputs 0. Else it outputs a random bit.

If $c$ has been generated by encrypting $m_b$ [i.e. $b = 0$] and $A$ is able to invert, then we see that $A'$ gives correct answer. Otherwise, if $A$ is unable to invert, $A'$ has a probability of 1/2 being correct.

$$
\therefore \text{Pr}[^{\text{CPA}}_{\text{PIA}}(n) = 1 | b = 0] = \text{Pr}[\text{invert}_A | b = 0] + \frac{1}{2} \left( 1 - \text{Pr}[\text{invert}_A | b = 0] \right) = \epsilon(n) + \frac{1}{2} (1 - \epsilon(n)) = \frac{1}{2} (1 + \epsilon(n))
$$
Proof

If \( c \) has been generated by encrypting \( m \) (i.e. \( b=1 \)) by a key say \( k \), what is the probability that \( A' \) returns 1?

Note that \( c \) must be the ciphertext of the message \( m \) for some other value of the key, say \( k' \). So, when \((c,m)\) is being given to \( A \), the probability that \( c \) is actually the ciphertext of a randomly chosen \( m \) is at most \( 2^{-n} \cdot 2^{-\alpha} = 2^{-\alpha} \). Then \( A \) inverts and obtains \((k',m',r')\), and if \( f(k',m',r')=(c,m) \), then it returns 0. Now this wrong, as \( b=1 \).
Otherwise, invert does not take place and there is 1/2 probability of \( A' \) to return the correct bit.

Conclusion

\[
\text{:. Pr}[\text{Priv}_{\text{CPA}}(n)=1 | b=0] = \frac{1}{2} (1 - \text{Pr}[\text{Invert}_{A'} | b=1]) \geq \frac{1}{2} (1 - 2^{-\alpha})
\]

Combining, \( \text{Pr}[\text{Priv}_{\text{CPA}}(n)=1] \geq \frac{1}{2} \cdot \frac{1}{2} (1 + \varepsilon(n)) + \frac{1}{2} \cdot \frac{1}{2} (1 - 2^{-\alpha}) = \frac{1}{2} + \frac{\varepsilon(n)}{4} - \frac{1}{2^{\alpha+2}} \)

Thus the indistinguishability of the encryption scheme under the assumption of an eavesdropper is violated. Thus \( \varepsilon(n) \) must be negligible.