CSC 591, Homework 1

Fatema Olia - 200253671
folia@ncsu.edu

October 1, 2018

1. (a) If we have a perfectly secure scheme where the ciphertext space is greater than the message space, i.e. $|C| > |M|$ then the encryption of any message is not uniformly random over ciphertext space because there are some ciphertexts that will never be reached during encryption.

   For example, consider a scheme like one time pad where we copy the first bit and append it to the end of the ciphertext. So we will have $|C| = \{0,1\}^{n+1}$. When we encrypt a message in this scheme we will always get messages starting and ending with the same bit, thus those messages in the ciphertext space whose start and end is different will never be reached. Thus, $Pr_{k \leftarrow KeyGen}[Enc_k(m) = c] < \frac{1}{|C|}$

   (b) For one time pad to be perfectly secure, we must at least have as many keys as there are messages, i.e. $|K| \geq |M|$. We know that $|S| = 7$.

   i. $M = S$ and $K = \{0,1\}^3$. Thus, $|M| = 7$ and $|K| = 8$. Since $|K| > |M|$ the scheme is perfectly secure.

   ii. $M = \{0,1\}^3$ and $K = S$. Thus, $|M| = 8$ and $|K| = 7$. Since $|K| < |M|$ the scheme is not perfectly secure.

   iii. $M = S$ and $K = S$. Thus, $|M| = 7$ and $|K| = 7$. Since $|K| = |M|$ the scheme is perfectly secure.

2. $G\{0,1\}^n \rightarrow \{0,1\}^{p(n)}$ is a PRG. For any input of size n, G gives an output of size $p(n)$

   (a) $G'(s) = s||G(s)$, where G is the pseudorandom generator.

   Intuitively, if a distinguisher D is given a string generated by G' then the distinguisher can take the first n bits of the string and generates an output using G, then compare the output to the rest of the string and come to know if the output comes from a truly random function or G'.

   Consider a probabilistic polynomial time distinguisher D that will try to guess if an input string is from G' or not. The algorithm for this distinguisher is as follows:

10

i. Input y to $D(y)$
ii. Parse $y = y_n||y_{p(n)}$
iii. Calculate $x = G(y_n)$
iv. IF $x = y_{p(n)}$ then Output 1
else Output 0

Thus, $Pr[D(y) \rightarrow 1] = 1$ (When $y = G'(s)$)
and $Pr[D(y) \rightarrow 1] = \frac{1}{2^n}$ (When y is truly random)
The difference between the two probabilities is non negligible. Thus, G' is not a pseudorandom generator.

(b) $G'(s) = f(G(f(s)))$, where G is the pseudorandom generator.

Intuition:
We can see the $G(f(s))$ is pseudorandom because G is a PRG. So even $f(G(f(s)))$ will be pseudorandom because it just removes the least significant bit of G

Thus, we shall now prove that if G is a PRG then G' is also a secure PRG.
Consider $\exists$ a probabilistic polynomial time adversary D that distinguishes the output of G.
Assumptions:
(1) G is a PRG.
(2) Towards a contradiction, assume $\exists$ a probabilistic polynomial time distinguisher D_1 that distinguishes the output of G' with a non negligible probability of p.
Observations:

20

- D_1 expects an input of $p(n) - 1$ bits where the string comes from G' or is a truly random string.
- D expects an input of n bits where the string comes from G or is a truly random string.

Reduction:
D receives an input X where X is either the output of G or a truly random string. In this case, if X is coming from G, it is generated by taking an $n+1$ bit string s and calculating $X \leftarrow G(f(s))$. ($f(s)$ is a function that removes the least significant bit of s)
Thus the algorithm for D is as follows:
$D(X)$

Good!

i. Compute $Z = f(X)$. ✓
ii. Send Z to D_1.
iii. When D_1 outputs a bit b_1, also output b_1.

Analysis:

- IF X is the output of G, Z is distributed exactly as the output of G'
Thus, $\Pr[D(X) \rightarrow 1] = \Pr[D_1(Z) \rightarrow 1]$ ✓
(Where $X \leftarrow G$ and $Z \leftarrow G'$)

- IF X is a truly random string, Z is distributed exactly as completely random
  Thus, $\Pr[D(X) \to 1] = \Pr[D_1(Z) \to 1]$
  (Where $X \leftarrow \{0,1\}^{n-1}$ and $Z \leftarrow \{0,1\}^{n-1}$)

However, as we have assumed in (2), D_1 distinguishes the output of G' with non negligible probability. From the analysis, this would mean that even D distinguishes G with non negligible probability. This is impossible since G is a PRG. Thus, our assumption must be wrong. Therefore, G' is also a pseudorandom generator.

3. (a) $F : \{0,1\}^n X \{0,1\}^n \to \{0,1\}^n$ and $F' : \{0,1\}^{2n} X \{0,1\}^{2n} \to \{0,1\}^{2n}$
$F'(k,x) = F(k_1,x_1)||F(k_2,x_2)$, where $k = k_1||k_2$ and $x = x_1||x_2$

F' is not a pseudorandom function as it can be distinguished by a disntinguisher with a non negligible probability. Consider the distinguisher D as follows:

Distinguisher:

i. query x_1 and obtain y_1

ii. query x_2, where the last n bits of x_2 are the same as the last n bits of x_1, and obtain y_2

iii. IF last n bits of y_1 match the last n bits of y_2, output 1
ELSE output 0

Analysis:
The distinguisher D has to guess if the output y_i is from the function F' or the truly random function T
$Pr[D^{F'} \to 1] = 1$
$Pr[D^T \to 1] = \frac{1}{2^{\frac{n}{2}}}$
Thus, the difference between the two probabilities is non negligible and so the distinguisher wins.
Hence F' is not a pseudorandom function.

(b) $F : \{0,1\}^n X \{0,1\}^n \to \{0,1\}^n$ and $F' : \{0,1\}^{2n} X \{0,1\}^n \to \{0,1\}^n$
$F'(k,x) = F(k_1,x) \oplus k_2$, where $k = k_1||k_2$
We can prove that F' is a pseudorandom function using reduction.

Assumptions:
(1) F is a PRF.
(2) Towards a contradiction, assume $\exists$ a probabilistic polynomial time distinguisher D_1 that distinguishes the output of F' with non negligible probability.
Observations:

- D_1 expects an input of n bits where the string comes from F' or from a truly random function.
- D is a distinguisher that expects an input of n bits where the string comes from F or is a truly random function.

Reduction:
$D(k, x)$
- Compute $y = F(k_1, x)$
- Send $z = y \oplus k_2$ to D_1
- When D_1 outputs a bit b, then also output b.

Analysis:

- IF y is the output of F, z is distributed exactly as the output of F'.
  $Pr[D_{y \leftarrow F} \rightarrow 1] = Pr[D_{1z \leftarrow F'} \rightarrow 1]$
- IF y is the output of a truly random function, z is truly random.
  $Pr[D_{y \leftarrow T} \rightarrow 1] = Pr[D_{1z \leftarrow T} \rightarrow 1]$

But the absolute difference $|Pr[D_{1z \leftarrow F'} \rightarrow 1] - Pr[D_{1z \leftarrow T} \rightarrow 1]|$ is non negligible according to our assumption. This means the difference $|Pr[D_{y \leftarrow F} \rightarrow 1] - Pr[D_{y \leftarrow T} \rightarrow 1]|$ would also be non negligible. However, since F is a pseudorandom function, this difference has to be negligible. Thus our assumption must be false and there cannot be a distinguisher D_1 that can distinguish F'.
Thus, F' is a pseudorandom function.

4. (a) **Intuition:**
If we have two schemes $\Pi_1 = (Gen_1, Enc_1, Dec_1)$ and $\Pi_2 = (Gen_2, Enc_2, Dec_2)$ and we generate a scheme $\Pi = \Pi_2(\Pi_1(m))$ (the message m is first encrypted by Π_1 and then the cyphertext generated by Π_1 is encrypted by Π_2) then:

- If Π_1 is CPA secure, then even if we decrypt Π_2 then we get c_1 (which is $\Pi_1(m)$) which is CPA secure and so Π is CPA secure.
- If Π_2 is CPA secure, then whether Π_1 is CPA secure or not we cannot decrypt Π_2 and so Π is CPA secure.

We can prove each case as follows:

i. **Assumption:**
(1) Π_1 is CPA secure
(2) Towards a contradiction assume there is an adversary A_1 that determines with absolute certainty which message Π is encrypting (m_0 or m_1)
Reduction:
When A_1 asks $Enc(m_i)$ do as follows:
- Send m_i to Π_2 which sends message m_i to Π_1 which encrypts the message and sends c_{i1} to Π_2 which encrypts it and returns c_{i2}.
- Output c_{i2} as c_i
When A_1 challenges with pair m_0, m_1, Do as follows:
- Pick a bit b.
- Send m_b to Π_2 which sends message m_b to Π_1 which encrypts the message and sends c_1^* to Π_2 which encrypts it and returns c_2^*.
- Output c_2^* as c^*
IF c^* matches c_0 output 0
else IF c^* matches c_1 output 1

Analysis:
If A_1 can match c^* to c_b, that means the output of Π_2 is deterministic, which means the output of Π_1 is deterministic.
Since Π_1 is CPA secure, it cannot have a deterministic output. Thus our assumption was wrong. The adversary A_1 cannot win the game if Π_1 is CPA secure.

ii. **Assumption:**
(1) Π_2 is CPA secure
(2) Towards a contradiction assume there is an adversary A_2 that determines with absolute certainty which message Π is encrypting (m_0 or m_1)
Reduction:
When A_2 asks $Enc(m_i)$ do as follows:
- Send m_i to Π_2 which sends message m_i to Π_1 which encrypts the message and sends c_{i1} to Π_2 which encrypts it and returns c_{i2}.
- Output c_{i2} as c_i
When A_2 challenges with pair m_0, m_1, Do as follows:
- Pick a bit b.
- Send m_b to Π_2 which sends message m_b to Π_1 which encrypts the message and sends c_1^* to Π_2 which encrypts it and returns c_2^*.
- Output c_2^* as c^*
IF c^* matches c_0 output 0
else IF c^* matches c_1 output 1
Analysis:
If A_2 can match c^* to c_b, that means the output of Π_2 is deterministic.
Since Π_2 is CPA secure, it cannot have a deterministic output. Thus our assumption was wrong. The adversary A_2 cannot win the game if Π_2 is CPA secure.

Hence, when $\Pi = \Pi_2(\Pi_1(m))$ then Π is CPA secure if at least one of Π_1 or Π_2 is CPA secure.