Cryptography – Ex02.

Oz Levi 305181158

Question 1.

a. Let assume that G' isnt a PRG, then we got a discern D' such that $D'(G'(x), U_{|X+1|}) >$ "½ + Neg".
With this D' we can build a D such that $D(G(x), U_{l(x)}) = D'(G'(x), U_{|x+1|}) / ½^{l(x)-|x|-1}$
Explanation: D will distinguish the n+1 prefix by the rule of D', and the other l(x)-|x|-1 bit, he is choosing randomalicly.
We got that...
$D(G(x), U_{l(x)}) = D'(G'(x), U_{|x+1|}) / ½^{l(x)-|x|-1}$
$D'(G'(x), U_{|x+1|}) / ½^{l(x)-|x|-1} >$ ("½ + Neg") / $½^{l(x)-|x|-1}$
("½ + Neg") / $½^{l(x)-|x|-1}$:= ("½ + Neg") / "Neg"
("½ + Neg") / "Neg" > "Neg".
We got a contradiction for G being PRG.

b. Lets build an discern D thats succed to distinguish between a output from $G_2(x)$ to some choose from $U_{|x|}$.
Define D: for some bit string y such that |y| = n+1, D caculate xor ofthe n-th first bits. If the solution equals to y[n+1], then D decide the y was taken from G_2, else taken from U_{n+1}.

c. Notice, from defenition, there is no existing discerns D such that $D(G(x), U_{l(x)}) > ½ +$ "Neg".
Here, G_3 making the same as G, except from when dealing with $x = 0^{|X|}$.
Lets assume that G_3 is not a PRG, so we got a D such that $D(G_3(x), U_{l(x)}) > ½ +$ "Neg", follow that because G_3 is the same for every x that diffrent from 0^n. (so $½^{l(n)}$ will be the case that $x=0^n$)
$D(G(x), U_{|x|}) > D(G_3(x), U_{l(x)}) + ½^{l(n)} > ½ +$ "Neg" $+ ½^{l(n)} = ½ +$ "Neg"
And this is a contradiction for G being PRG.

d. G_4 is not a PRG, a discerns D can use G like this:
For some x $D(G_4(x),U_{|x|})$ is from G_4 if half the output c is the same as $G(0^{|x|})$.
Then D got advantage by discern between $(G_4(x),U_{|x|})$.
Lets calculate the advantage: if the output is really from G_4 then we will succed by probability 1. By the assumption that our n/2 prefix of the output is consistent the probability for not been generated from G_4 is to guess all the prefix, so $½^{n/2}$.
So the probability to succes is $1-½^{n/2}$ that is greater then "Neg", and G_4 is not a PRG.

Question 2.

We will define G'_i as G' that runs only l(x)-i-1 loops (from 1+i to l(x)), notice that G'_i still generating output thats l(|x|) long.
the other bit for the output will be generated by the following:

if j <= i, output[j] = randomicly.

else, will generate the bit as followed by G'.

We will define D_i as follow: $D_i = D(G'_i(x,l(|x|)),G'_{i+1}(x,l(|x|)))$

Lets assume that there is an attacker E with polynomic function l(|x|) that his distinguishability for our $G'_{l(|x|)}$ is greater then ½ + "Neg".
That mean, there is existing D such that $D(G'_{l(|x|)}(x,l(|x|)),U_{l(|x|)}) > ½ + $ "Neg"
~~Now we will notice that for every D_i , D_{i+1} it follows $D_i - D_{i+1}$ >= ½ (one bit changed).~~
Next by hybrid argument, $D(G'_0(x,l(|x|)),G'_{l(x)}(x,l(|x|)))$ is also ½ + "Neg", but
$G'_{l(x)}(x,l(|x|)) = U_{l(x)}$
So $D(G'_0(x,l(|x|)), U_{l(x)}) = ½ +$ "Neg".
With this information, we can build a new D' such that will distiguish between an output from G and G'_0.
D' will decide if the output is from G or G'_0 by taking the input send it to the Oracle (using the Oracle that knows to generate same as G). If the result is same our output, then we know that it was generate from G, else from G'_0.
This D' succed with probability 1 and working polynomicly (using G is polymonicly bt defenotion).
By combining those D and D' we can succed to build a discern that knows to distinguish between an output from G or from $U_{l(x)}$ and by choosing l(x)=|x|+1 then we found a discern between G and $U_{|x|+1}$.
And thats contractiction to the fact that G is PRG.

~~For l(|x|) = |x| + 1, we will see thats~~
~~$D(G'_0(x,l(|x|)),G'_{l(x)}(x,l(|x|))) = D(G'(x,|x|+1)),G'_{|x|+1}(x,|x|+1))$~~
~~$D(G'(x,|x|+1)),G'_{|x|+1}(x,|x|+1)) = D(G'(x,|x|+1)),G(x))$~~
~~And thats contractiction to the fact that G is PRG.~~
~~???~~
~~he is generating with the original G. define D_i as the hybrid probabilty of G'_i (need to save the first generate). For every 0 < i < l, $D_i - D_{i-1}$ < neg. So every two consecutive D's are indistinghisheale. We will see that $D_0 = G'$ and $D_l = G$. We alreay know that G and U_l are indistinguishable, and shawn by hybrid that G and G' are indistinguishable, so done.~~

- ~~Maybe G_i doinf the for i times, and the other bits flip cion.~~
- ~~Orrr return the real x till index n+1, and the others flip coin.~~

Question 3.

Lets assume exictence of PPT attacker called E thats winning the CPA game. E's using the Oracle for Vector of t messages.

Oracle: using Enc(r, m_i xor F_k(r)) function for some function F_k that stay the same , and r that randomized for every use. The Oracle retrun the c_i answare from Enc() function.

If the Oracle had used the same r for two differents messages m_i, m_j, E can make xor between c_i, c_j and get information about the messages and the function. Then by our assumption E does this with probability greater then "½ + Neg" when the F_k that the Oracle use is random function. But if the Oracle using a pseudo random function as F_k the chance for attacker E winning the game is smaller then "½ + Neg".

Lets build attacker E': E' choosing some function F (can be random or pseudo random), E' is using same technic as E and deciding that the Oracle will use the chosen F for Enc(). The b (chosen bit) answare for E returns to E'. E' checking E's winning. if E has win that game (that by assumtion we got greater then "½ + Neg" probability) that means that our E' chose a random function F from the world of all function.

For the reasone that E' is polynic, we got a contradition fot the defenition of pseudo-random functions.

NOTICE! We are refering to function from $\{0,1\}^n$ to $\{0,1\}^n$.