MIDN ANDRES
SY308 - 4001
HW04

HW Questions

1. (10 pts) Read the course notes through Encryption w/ Python. Honestly declare one of the following:
   a. (0 pt) I didn't read the notes.
   b. (2 pts) I just skimmed the notes.
   c. (3 pts) I read the notes, but I skipped a couple of difficult topics.
   [✓ d.] (5 pts) I read the notes very carefully and thoroughly.

D

2. (5 pts) Solve the quiz candidate problems in the notes through Perfect Indistinguishability. Honestly declare one of the following:
   a. (0 pt) I did nothing.
   b. (2 pts) I looked at the solutions, but didn't attempt to solve the problems on my own. (Recall you should get help from your instructor, if you don't understand the course material.)
   [✓ c.] (4 pts) I tried to solve **some** of the problems on my own, and then checked the solutions.
   d. (5 pts) I tried to solve **all** problems on my own, and then checked the solutions.

C

3. (15 pts: 20 minutes) Consider the private-key encryption scheme. We saw this in this previous lecture. We will see that it does not provide IND-CPA security, although it's indistinguishable against an eavesdropping adversary (i.e., under a ciphertext only attack).

Let G be a pseudorandom generator with expansion factor $\ell()$. Consider thefollowing encryption scheme with message space $\mathcal{M} = \{0,1\}^{\ell(n)}$.

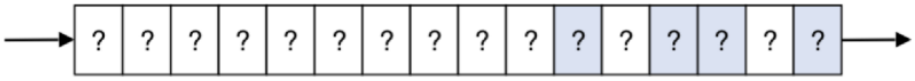
- Gen(n): Choose $K \leftarrow \{0,1\}^n$ uniformly at random and output K.
- Enc$_K(M)$: Output $G(K) \oplus M$. Here $m \in \{0,1\}^{\ell(n)}$.
- Dec$_k(C)$: Output $G(K) \oplus C$

You, as an adversary, can feed any pair messages (M0,M1) you want, and the box will give you a ciphertext C encrypting Mb, where b is hidden to you. You must figure out whether this hidden b is 0 or 1.

1. Figure out whether b is 0 or 1, or equivalently whether C.bin is the encryption of M0.txt or M1.txt using the chosen-plaintext attack (i.e., choose some pair messages and feed them to the box and get a ciphertext encrypting one of the two from the box).
2. Give a detailed explanation how you figured out the answer.
3. In addition, save the execution log into log1.txt that supports your argument.

In order to carry out a chosen-plaintext attack I created two messages, message 0 consisted of hundreds of "A"s, message 1 consisted of a single "A." The encryption scheme makes it clear that the length of the message and the length of the key will be equivalent, thus the ciphertext will correspond to the length of the message with padding. With this knowledge I will be able to determine whether b is 0 or 1. First I sent message 0 then message 1, the resulting ciphertext was 16 bytes, under this assumption, message 1 was encrypted and message 0 was discarded. To prove this, I sent message 1 then message 0, the resulting ciphertext was 100s of bytes long thus confirming my assumption and proving that b is a 1.

4. (10 pts: 15 minutes) We know that LFSRs are not good PRGs when used on their own. In this problem you will demonstrate it. Consider the following LFSR with a 16-bit internal array, taps at the locations marked in blue.



Suppose that you do not know the initial seed, but you see the following output from this LFSR:

100100011000110001

This is in order of output, meaning that the first bit that came out was 1, then 0, then 0, then 1, then 0, etc.

- (5 pts): Determine the original seed that was used in the LFSR based on this output. Hint: think about where the first output bits come from when an LFSR starts.
- (5 pts): Predict the next 4 bits that will be output by this LFSR.

100100011000110001

OG -1 = 00100011000110001?

$? \oplus 0 \oplus 0 \oplus 1 = 1$

$? = Ø$

OG = 001000110001100010 → Ø

Given = 100100011000110001 → 1

$1 \oplus 0 \oplus 0 \oplus 1 = Ø$

1 = 010010001100011000 → Ø

$0 \oplus 0 \oplus 1 \oplus 0 = 1$

2 = 101001000110001100 → Ø

$0 \oplus 1 \oplus 1 \oplus 0 = Ø$

3 = 010100100011000110 → Ø

$0 \oplus 0 \oplus 1 \oplus 0 = 1$

4 = 101010010001100011 → 1