CS 4732/57322 Homework #3

Due electronically by midnight 7/12/2018. There will be a one letter grade penalty for every day late, until 3 days after this date. No homework will be accepted after that time.

For submission, if done on paper please scan and submit as a pdf. If done in word, please submit the .docx or .doc format.

IMPORTANT: Clearly indicate outside resources utilized and sign below. Failure to cite use of outside resources will be reported for appropriate disciplinary actions. Note that discussions with other students are encouraged; copying – with or without modifications – is unacceptable and will also be reported.

I discussed one or more problems with the following people:

I hereby certify that any outside resources utilized, other than the textbook and class materials, are clearly cited. All other material I provide for this homework submission is my own original work.

- Course Powerpoints and Book

Thomas Mintun

Printed name

1. (8 points) In regards to whether or not a sequence is random, two criteria would be uniform distribution and independence. Define the two, then give me an example of a sequence (at least 10 bits long) that has a uniform distribution but violates independence. Relate to me the independence.

In regards to whether or not a sequence is random, two criteria would be uniform distribution and independence. Uniform distribution refers to the sequence of 0's and 1's being uniform for a random bit sequence. This is because mathematically random behavior would generate 50% 0's and 50% 1's as in flipping a fair coin a significant number of times, and a random sequence should mimic this. Independence refers to a property that no one bit in a sequence can be inferred by another bit in the sequence. If one bit or character or integer could be inferred in a sequence of random numbers than that sequence would not be random. A sequence of bits that has uniform distribution but violates independence is: 0101010101010101010101... ---> This sequence has eleven 0's and eleven 1's i.e. uniform distribution, but violates independence because every 0 is followed by a 1. Every 1 is followed by a 0 so the next bit can always be inferred.

2. (8 points) Why is it a bad idea to reuse a stream cipher key (in this case, the stream cipher is just exclusive or'ing your key with the plaintext)? Explain with an example using two plaintext bitstreams of 8 bits involving the reuse of a key bitstream of also 8 bits. Then XOR the resulting two ciphertexts. What property does it have? This might be hard to see, but look at the two plaintexts in relation to it.

Darths (attackers) will try to get you to encrypt a bunch of 0's. XORing a bunch of 0's with a binary key will result in the key being the ciphertext because of XOR's properties. But the reason why it's a bad idea to reuse a stream cipher key is because encrypting two different plaintext messages with the same key for a stream cipher allows an attacker to XOR the two resulting ciphertext messages removing the information about the key. The result can be used to reconstruct the plaintext messages. If the bit in CT1 XOR CT2 is a 0 you know both PT1 and PT2 had a 0 or 1 in that bit, and if a 1 is in that bit position then you know PT1 has a 1 in that position while PT2 has a 0 in that position or opposite.

PT1: 1010110000	PT2: 1111110000	CT1: 0000011010
XOR Key: 1010101010	XOR Key: 1010101010	XOR CT2: 0101011010
=CT1: 0000011010	=CT2: 0101011010	= 0101000000 → this is PT1 XOR PT2!!!!!!

3. (12 points) While it is desirable to have a PRNG have a full period, that does not guarantee good randomness. Consider a linear congruential generator using the settings of a = 6, m = 13 and another one with a = 7, m = 13. Write the full two sequences out for their periods. Which one appears more "random"? In other words, if you had to pick one, which one would you use?

Seed $X_0 = 1$ C = 0 ||||| $X_{i+1} = (a*X_i + C) \bmod M$

(6*1 + 0) mod 13 = 6; 6*6 mod 13 = 10; 10*6 mod 13 = 8……..

7*1 mod 13 = 7; 7*7 mod 13 =

Period 1 (a=6, m=13): 6,10,8,9,2,12,7,3,5,4,11,1,6 Period = 13

Period 2 (a=7, m=13): 7,10,5,9,11,12,6,3,8,4,2,1,7 Period = 13

The more random one is the second one. The higher a value will give more entropy. The higher value a in class example gave more entropy. I would change the seed if doing question over again to get more entropy.

4. (10 points) What is the difference between condition testing and health testing in regards to TRNGs. Give an example of a bitstream that would pass health testing but not conditioning and explain why.

A true random number generator may produce output that is biased in some way (more ones than zeros or vice versus). Condition testing tests for this bias, and conditioning algorithms modify a bit stream to further randomize the bits and increase entropy. So a TRNG produces random bits, and conditioning is done to make sure those random numbers are in the proper range and not favoring something that the TRNG is temporarily favoring. For instance a TRNG reading temperature may provide too many consistent temperatures that are the same if the thermometer was placed in North Pole where temperature does not vary much. So a conditioning algorithm would condition those random numbers to be in a specified range or just take the least significant bit of the reading every second. Health tests determine if the noise source is performing as expected. Health tests are also placed on the conditioning component to assure that the output behaves as a true random bit stream.

5. (8 points) Is it likely that eventually asymmetric key encryption will negate the need for symmetric key encryption? Make an argument for why this is or is not the case.

No! Asymmetric key encryption is very slow, and is vulnerable to the same attacks as symmetric key encryption. Symmetric key encryption should be used for a significant or large amount of data. Also, asymmetric key encryption is NOT more secure than symmetric key encryption. In practice, asymmetric key encryption is used to give someone else your/the AES (symmetric) key because communicating via symmetric key encryption is much faster. This prevents the need for a central key exchange that could be hacked also.

6. (10 points) You intercept a ciphertext message 9 sent to a user using RSA. The public key is e=5 and n=35. What is the plaintext number? Show all your work.

$C = M^e \bmod N$

$M = C^d \bmod N = (M^C) \bmod N = M^{ed} \bmod N$

d = 5 chosen

So $M = 9^5 \bmod 35 = 59049 \bmod 35 = 4$