CS 4920/5920 Applied Cryptography Spring 2018

Assignment 4

This assignment is due at the beginning of class on March 19. Problems 4 and 5 are computer problems.

Problem 1.

With the ECB mode, if there is an error in a block of the transmitted ciphertext, only the corresponding plaintext block is affected. However, in the CBC mode, this error propagates. For example, an error in the transmitted C_1 (Figure 7.4) obviously corrupts P_1 and P_2.

a. Are any blocks beyond P_2 affected?
b. Suppose that there is a bit error in the source version of P_1. Through how many ciphertext blocks is this error propagated? What is the effect at the receiver?

Problem 2.

Consider the linear congruential algorithm with an additive component of 0, i.e.,

$$X_{n+1} = (aX_n) \bmod m$$

a. It can be shown that if m is prime and if a given value of a produces the maximum period of m-1, then a^k will also produce the maximum period, provided that k is less than m and that k and m-1 are relatively prime. Demonstrate this by using X_0=1 and m=31 and producing the sequences for $a^k = 3, 3^2, 3^3$, and 3^4.

Now consider $m=2^4$, i.e., $X_{n+1} = (aX_n) \bmod 2^4$. (Hint: This is a common case for a linear congruential algorithm when m is a power of 2.)

b. What is the maximum period obtainable from the generator with $m=2^4$?
c. What should be the value of a to achieve the maximum period?
d. What restrictions are required on the seed?

Problem 3.

RC4 has a secret internal state which is a permutation of all the possible values of the vector S and the two indices *i* and *j*.

a. Using a straightforward scheme to store the internal state, how many bits are used?
b. Suppose we think of it from the point of view of how much information is represented by the state (information entropy). In that case, we need to determine how many different states there are, then take the log to base 2 to find out how many bits of information this represents. Using this approach, how many bits would be needed to represent the state?

Problem 4.

Let's implement AES.

a. Identify the software program and the programming language you use for this problem.
b. Attach a part of your code/implementation that corresponds to the MixColumns computation. Include comments and describe the variables/functions (e.g., the MixColumns matrix [02 03 01 01; 01 02 03 01; …] that you built previously).
c. Perform Key Expansion using the Key {0f1571c947d9e8591cb7add6af7f6798} and construct a table that is similar to Table 6.3 (including the elements in Table 6.3).
d. Perform AES on Plaintext {0123456789abcdeffedcba9876543210} using the key in Problem 3.c. and construct a table that is similar to Table 6.4 (including the elements in Table 6.4).

Problem 5.

Let's study the Avalanche Effect in AES.

a. Explain the Avalanche Effect and why it is an attractive feature.
b. Given the Plaintext *P*= {0123456789abcdeffedcba9876543210} as the baseline, generate three additional plaintexts that are different from *P* by one bit and perform AES on them using the key {0f1571c947d9e8591cb7add6af7f6798}. To compare each of the new plaintexts and *P*, construct three tables that are similar to Table 6.5. (If you have not used your own implementation from Problem 3, identify what you used for this problem.)