Problem 1 [10pts]

An encryption scheme is said to be “deterministic” if encryption of the same plaintext value always results in the same ciphertext value. In other words, first time one encrypts P, the resulting ciphertext is C; the next time one encrypts P, the resulting ciphertext is still C, and so on.

Is a deterministic encryption scheme secure against the “Known-Plaintext Attack”, we discussed in class? Explain why or why not. Give an example to illustrate your answer.

Problem 2 [15pts]

In the class, we studied the Caesar’s cipher. A similar cipher is called affine cipher which works as follows.

Encryption: each plaintext letter P is encrypted to obtain the ciphertext letter C such that C = aP + b (mod 26), (where a,b are numbers between 0,1,….,25, and represent the secret key).

Decryption: each ciphertext letter C is decrypted to obtain the plaintext letter such that P = (C-b)a⁻¹ (mod 26).

I need to send a message to the class “HELLOCLASSALLOFYOULLGETANAYOUNOTNEEDTODOTHEHOMEWORK” (I don’t mean it!), and want to send it encrypted using the affine cipher so that the department chair does not learn the message ☻. The chair intercepts the 8th and 3rd letters of the cipher text, ‘J’ and ‘Q’ respectively, and somehow learns the corresponding plaintext letters, i.e., ‘A’ and ‘L’ respectively. Can he decrypt the message (and take a disciplinary action against me ☻)? If so, explain how? What is the secret key? What is the original ciphertext that I sent out?

Problem 3 [25pts]

A 64-bit long message “10101000 01010101 01001010 10101001 10101001 10101011 11010101 11111010” (ignore the “spaces”) was encrypted with DES using the key “1011010 0101101 0111101 0101101 0110101 0110101 0110100 1101101” and the
following ciphertext was obtained “10010101 01101010 10111101 10001001 10101111 01010101 10101010 10001001”

Can you figure out the ciphertext when the message “01001011 10101010 10110101 01010110 01010110 01010110 10101010 00000101” is encrypted with DES using the same key used in the previous operation “1011010 0101101 0111101 0101101 0110101 0110101 0110100 1101101”? If so, explain the details how. If not, explain the reason why.

[Hint: You don’t need to write a program (in fact, the values given above are all imaginary – when you encrypt the given message with the given key, you WILL NOT get the given ciphertext. You only need to know how DES works and look at its operations (which was explained in the class)]

**Problem 4 [30pts]**

The reading assignment in the last lecture covered how AES (Rjindael) works. This problem will give you an idea as to how much processing time AES requires to perform key generation, encryption and decryption.

Download the AES implementation source code from: [http://islab.oregonstate.edu/koc/ece575/02Project/Sha/sprash.zip](http://islab.oregonstate.edu/koc/ece575/02Project/Sha/sprash.zip)

First, familiarize yourself with the AES key generation, encryption and decryption functions in this source code. Choose a key and block size of 128 bits. Choose any plaintext M 128*5 bit long. Then, perform the following:

1) Execute the key generation function to get a key K
   • Compute the execution time.

2) Execute the encryption function to encrypt M using K, and obtain the ciphertext
   • Compute the execution time.

3) Execute the decryption function to decrypt C using K, and to obtain M back.
   • Compute the execution time.

Repeat each of the above steps 100 times and give the average execution time for each of the three functions. List the type and speed of the processor, and the memory (RAM) of the machine you execute the code on.

I hope you know how to measure execution time! If not, please try to figure it out yourself before asking for help. Include a description of how you measured the timing, such as what functions you measured the execution time of.

**Problem 5 [10 + 10 = 20pts]**

1. A 7-bit long message (assume that the message space is $2^7$) was encrypted using one-time pad to yield a cipher-text “0110011”. What is the probability that the corresponding plain-text was “1000001”. Explain your answer.

2. A message m1 was encrypted with a key K using one-time pad and the ciphertext was transmitted. The encryptor became lazy and encrypted another message m2 with the
same key K using one-time pad, and the ciphertext was transmitted. Is this a secure way of encryption? Why/why not?