CS 5460: Computer Security I
Fall 2019
Assignment 1: Build a Symmetric Key Cryptosystem Using Multiple Encryption/Decryption Techniques

Total Marks: 100

In this assignment, you need to write a computer programming code to build a cryptosystem that accommodates multiple steps, each representing a distinct cryptography technique. So, the system needs a composite key - multiple keys placed adjacent to the next - one for each crypto technique. In this assignment, the composite key consists of a set of numbers, where each number has two digits (there will be no space or separator between the numbers or digits, when they will be given as input to your program). The last number of the key represents the ‘Key’ for one-time pad crypto technique, and the previous numbers are used to generate a ‘Key’ for Columnar Transposition crypto technique by using a Polybius square.

Encryption Process:

Input: i) Composite Key, ii) Plaintext Message (non case-sensitive)

Output: Ciphertext Message

Let's assume that the Composite Key is: 1422555515

Here, the key for one-time pad is 15 (last number consisting of two digits). The previous part of the composite key: 14225555 will be used to generate a key for columnar transposition, using a polybius square, where each number consists of two digits. So, we have the following numbers: 14, 22, 55, 55. Now, we will use the following polybius square (fixed for this assignment) to get the key for columnar transposition, which in this case, would be: BALL

	0	**1**	**2**	**3**	**4**	**5**
0	E	Y	O	P	D	9
1	2	H	Q	X	1	I
2	R	3	A	J	8	S
3	F	0	N	4	G	5
4	Z	B	U	V	C	T
5	M	7	K	W	6	L

Table: Polybius Square

Now, we will use this key: BALL to encrypt the given plaintext message using Columnar Transposition (No Padding is needed).

Let’s note the output of Columnar Transposition as **cipher1.** In the second step, cipher1 will be encrypted using One-time Pad. To do so, use the given polybius square to find the numerical representation for each letter in cipher1 (like, you will get 43 for G), convert each number (consisting of two digits) to the corresponding 6-bit binary value (e.g., we will get a binary value: 001011 for the decimal number: 11), and then implement one-time pad crypto technique upon converting the key to its binary value (in this example, ‘15’ would be converted to ‘001111’). Implement one-time pad crypto technique individually for each letter in cipher. Then convert each 6-bit binary number to the corresponding decimal number, which would represent the final output of the encryption process.

Let’s assume, cipher1 is: ERFZM. So, the polybus square would give us the following numbers: 00, 02, 03, 04, 05. The binary representation would be: 000000, 000010, 000011, 000100, 000101. Using key: 15 (001111) for one-time pad encryption, we get the following ciphertext for ERFZM by performing XOR operation: 001111, 001101, 001100, 001011, 001010. So, the corresponding decimal numbers would be: 15, 13, 12, 11, 10. Thus, if cipher1 is ERFZM, the final ciphertext would be: 1513121110

Task 1 [Marks: 50]
Given a plaintext and a composite key, generate the ciphertext using encryption process described above.

Task 2 [Marks: 50]
Given a ciphertext and a composite key, retrieve the plaintext using decryption process, which should be based on the encryption process described above.

In the expected scenario, if the final output (ciphertext) of Task 1 is given as an input to Task 2, it should retrieve the plaintext given as an input to Task1 (assumption: The same composite key is used in Task 1 and Task 2).

Instructions:

- You will need to submit a working version of your code through email to GTA of this course (Manazir Ahsan, email: manazir.ahsan@aggiemail.usu.edu) before **11:59PM on Sunday, September 15**. If needed, add additional instructions for running the code in a ‘Read Me’ file.

- Your program needs to have input fields, required for encryption and decryption process described above, and mechanism to show the output. The program without required input fields and visible output will not be acceptable.

- One submission is required from each group (all group members need to be cc'd in the submission email). See Late Submission Policy in course syllabus.

- The subject-line of the email: **CS 5460: Assignment 1 Submission: <Group Name>**

- Each group, with all members present, will need to demonstrate the code on **Monday, September 16**. Attendance during demonstration is required. A student will not receive the grade for the assignment if he/she fails to attend the demonstration.