Tutorial #2
Security in Computing COSC2356/2357

Q1. a) Use exclusive or (XOR) to "add" the bit strings 11001010 ⊕ 10011010.

b) Convert the decimal numbers 8734 and 5177 into binary numbers, combine them using XOR, and convert the result back into a decimal number.

Q2. [Do It Yourself] Using **One-Time Pad** encryption algorithm, encrypt the plaintext "**MINIMUM**". The **key** is: **101 110 111 001 000 010 111**. Again, decrypt the ciphertext using the same key.

[*Hints:* Convert each alphabet of the plaintext in to binary using the following dictionary (Table-Q3).]

Table-Q3: Dictionary

A	I	M	N	U	X	V	Y
000	001	010	011	100	101	110	111

Q3. A stream cipher can be viewed as a generalization of a one-time pad. Recall that the one-time pad is provably secure. Why can't we prove that a stream cipher is secure using the same argument that was used for the one-time pad?

Q4. Recall the online bid method discussed in the lecture

a) What property or properties of a secure hash function *h* does this scheme relies on to prevent cheating?

b) Suppose that Charlie is certain that Alice and Bob will both submit bids between $10,000 and $20,000. Describe a forward search attack that Charlie can use to determine Alice's bid and Bob's bid from their respective hash values.

c) Is the attack in part (**b)** a practical security concern?

d) How can the bidding procedure be modified to prevent a forward search such as that in part (**b**)?

Task 1 (Symmetric Key Encryption and Decryption using OpenSSL).

It is assumed that you have OpenSSL installed in your computer. If you are using Microsoft Windows operating system, then download and install OpenSSl from the following link:

http://downloads.sourceforge.net/gnuwin32/openssl-0.9.8h-1-bin.zip

Unzip the file.

Note: If you are using linux or recent MacOS, then you already have OpenSSL.

Locate the "**openssl.exe**". Assume that the "**openssl.exe**" file is in **D:\OpenSSL\bin**. Open terminal (command prompt in Windows operating system) and run the following command from the directory mentioned above:

```
>openssl
```

You are ready to run OpenSSL command.

Task 1.1 (AES Algorithm) Assume that you have a plain-text file, called "**`textFile.txt`**", with your *name* and *student ID* in that file. The may look like the followings:

Student ID: S1234567 Name: ABCDEF

Apply Openssl's *AES algorithm* using the followings:

ECB mode and a 256-bit key to encrypt and decrypt. Choose a reasonable shared secret key (e.g 1234). Also try for different key size (e.g. 128-bit).

Solution:

Step-1 (Encryption): Encrypt a text file called "**`textFile.txt`**" and generates a binary ciphertext file called "**`secret.txt`**" using the following command.

```
aes-256-ecb -in textFile.txt -out secret.txt
```

Now a password will be asked. Enter **1234** as password. Enter the same password to verify the previous password. This password will be required to decrypt the file.

Step-2 (Decryption): Decrypt the "**`secret.txt`**" file using the following command to obtain the plaintext file:

```
aes-256-ecb -d -in secret.txt -out decrypt.txt
```

Enter **1234** as password.

Check the file **`decrypt.txt`** where you should find the same content as in **`textFile.txt`**.

[**Do It Yourself**]

Task 1.2 (DES Algorithm) Assume that you have a plain-text file, called "**`textFile.txt`**", with your *name* and *student ID* in that file. Apply Openssl's *DES algorithm* using ECB mode to encrypt and decrypt. Choose a reasonable shared secret key.

Solution:

Step-1 (Encryption): Encrypt a text file called "**`textFile.txt`**" and generates a binary ciphertext file called "**`secret.txt`**" using the following command in the OpenSSL terminal.

```
des-ecb -in textFile.txt -out secret.txt
```

Now a password will be asked. Enter **1234** as password. Enter the same password to verify the previous password. This password will be required to decrypt the file.

Step-2 (Decryption): Decrypt the "**`secret.txt`**" file using the following command to obtain the plaintext file:

```
des-ecb -d -in secret.txt -out decrypt.txt
```

Enter **1234** as password.

Check the file **`decrypt.txt`** where you should find the same content as in **`textFile.txt`**.