SNS Assignment 2, Execution Document

This document contains the commands required to run the program for each of the 5 questions present in the Practicals 2 document

Roll Number: BT18CSE**046**

Question Set: B

Socket programming is done in C, specifically using winsock2 library (will work on windows only)

1. Symmetric Cryptosystem

- **Z** : Implement 2-round Feistel cipher. Suitably, choose your own function f, such that f uses your roll no last two digits in its computation. (2.5 M)
  - **Files provided** : BT18CSE046_SE_Z_Kg.c, BT18CSE046_SE_Z_En.c, BT18CSE046_SE_Z_De.c
  - **Compilation commands**:
    - Keygen program: gcc -o keygen BT18CSE046_SE_Z_Kg.c
    - Encryption program: gcc -o alice BT18CSE046_SE_Z_En.c -lws2_32
    - Decryption program: gcc -o bob BT18CSE046_SE_Z_De.c -lws2_32
  - **Execution Steps:**
    - Generate the required key using : ./keygen.exe <key_size>
      - Where the key_size for the feistal cipher is assumed to be the sum of the 2 round key sizes, k1 and k2 (k1+k2)
      - keys.txt will be created
    - Start the receiver program(decryptor) in listening mode: ./bob
    - Start the encryption program with command line input of message to be encrypted: ./alice <message>
      - Where message is the message to be encrypted and sent over sockets to the decryption program.
  - **Sample Run:**
    - Compiling and key generation

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                                   powershell

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal Cipher\Submission Files> gcc -o keygen BT18CSE046_SE_Z_Kg.c
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal Cipher\Submission Files> gcc -o bob BT18CSE046_SE_Z_De.c -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal Cipher\Submission Files> gcc -o alice BT18CSE046_SE_Z_En.c -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal Cipher\Submission Files> ./keygen 64
10100111111001011111101111011100
10000111000100100010110100111010
Done!
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal Cipher\Submission Files>
```

    - As apparent above, 2 32 bit keys have been created when input was 64 bit key(design choice for keygen)
    - Execution

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal C
ipher\Submission Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal C
ipher\Submission Files> ./bob
Socket created
```

Start decryption program in listening mode before running the encryption program

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal C
ipher\Submission Files> ./alice hello
Conn established!
Key Size:32
k1:10100111111001011111101111011100
k2:10000111000100100010110100111010
Plain Text:0000000000000000000000000011010000110010101101100011011
0001101111
Encrypted data: 11000010100010011010001111101101001000000111101111
101011010001110
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal C
ipher\Submission Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal C
ipher\Submission Files> ./bob
Socket created
Connection made
Alice sent(encrpyted): 11000010100010011010001111101101001000001
1110111110101101000111 0
Key Size:32
k1:10100111111001011111101111011100
k2:10000111000100100010110100111010
Decrpyted data:0000000000000000000000000011010000110010101101100001
10110001101111
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\Feistal C
ipher\Submission Files>
```

- As we can see in the above ss, the plain text before and after encryption match (bits output in ss)
- The input data was padded to match 64 bits

- **B**: DES CFB mode (2.5 m)
  - **Files Provided**: BT18CSE046_SE_B_Kg.c, BT18CSE046_SE_B_En.c, BT18CSE046_SE_B_De.c
  - **Compilation Instructions:**
    - Keygen: gcc -o keygen BT18CSE046_SE_B_Kg.c
    - Decryption Program: gcc -o bob BT18CSE046_SE_B_De.c -lws2_32
    - Encryption Program: gcc -o alice BT18CSE046_SE_B_En.c -lws2_32
  - **Execution steps:**
    - Keygen: ./keygen
    - Run decryption program: ./bob
    - Run encryptor with command line args: ./alice <message>
      - Where message is to be encrypted
  - **Sample run:**
    - Compilation and key generation:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                                  powershell

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submission Files> gcc -o keygen BT18CSE046_SE_B_Kg.c
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submission Files> gcc -o bob BT18CSE046_SE_B_De.c -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submission Files> gcc -o alice BT18CSE046_SE_B_En.c -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submission Files> ./keygen
Generating a random 64 bit key[bits will be dropped by functions]
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submission Files>
```

    - This generates the des 64 bit key(which will automatically be reduced to 56 by functions) in file called key.txt
    - Start the decryption program in listening mode

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submi
ssion Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submi
ssion Files> ./bob
Socket created
```

- Now finally run encryption program

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submi
ssion Files> ./alice "This is some long message to be encrypted"
Conn established!
Bits before encryption:0101010001101000011010010111001100100000001
1010010111001100100000011100110110111101101101011001010010000001 1
0110001101111011011100110011100100000011011010110010101011100110111
0011011000010110011101100101001000000111010001101111001000000011100
0100110010100100000011001010110111001100011011100100111100101110 0
0001110100011001010110010000000000000000000000000000000000000
Bits after encryption:0011010100110011101010001011011101000101110
0110111110100101110011000000010101000011000010010010110011011001 1
1000111110000000001011001111001111101011110101101001110011010110011
1000101001000111101000001100111000111110111001010010011000010100 0
1001101110011101100001000111101111010000110101101001101000111101 0
1101101010001000100010000011110011110001001111010001100100
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submi
ssion Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submi
ssion Files> ./bob
Socket created
Connection made
Alice sent(encrpyted): 0011010100110011101010001011011101000101110
0011011111010010111001100000001010100001100001001001011001101100 1
1100011111000000001011001111001111101011110101101001110011011011001
1100010100100011110100000110011100011111011100101001100001010 0
0100110111001110110000100011110111101000011010110100110100011110 1
0110110101000100010001000001111001111000100111101000110100

Before decryption:0011010100110011101010001011011101000101110110
1111101001011100110000001010100001100001001001011001101100111000
1111100000000101100111100111101011111010110100111001101011001100 0
1010010001111010000011001110001111011100101001100001010001001
1011100111011000010001111011110100001101011010011010001111010110 1
1010100010001000100000111100111100010011110100011010 0

Decrypted bits:0101010001101000011010010111001100100000011010010 1
1100110010000001110011011011110110110101100101001000000110100011
0111101101110011001110010000001101101011001010110110011011100110 1
0001011001110110010100100000011101000110111100100000011000100110 0
1010010000001100101011011100110001101110010011110010111000011101
0001100101011001000000000000000000000000000000000000

Decrypted text by Alice:This is some long message to be encrypted
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q1\DES\Submi
ssion Files>
```

- As we can see, the output is correctly decrypted

2. Asymmetric Cryptosystem

- **B**: Elgamal (5m)
  - **Files Provided**: BT18CSE046_AC_B_Kg.c, BT18CSE046_AC_B_De.c, BT18CSE046_AC_B_En.c
  - **Compilation Instructions**:
    - **Keygen**: gcc -o keygen BT18CSE046_AC_B_Kg.c -lgmp
    - Decryption Program: gcc -o bob BT18CSE046_AC_B_De.c -lgmp -lws2_32
    - Encryption Program: gcc -o alice BT18CSE046_AC_B_En.c -lgmp -lws2_32
  - **Execution steps**:
    - Generate keys: ./keygen <key_size>
      - Where key_size is the given as k, and the primes for elgamal will be chosen from 1^k (111…1 k times)
    - Run decryption program: ./bob
    - Run encryption program with command-line arguments: ./alice <message>
      - Where message is and integer base-10 from 1 to chosen prime q-1
  - **Sample run:**

- Compiling and keygen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                                                    powershell

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submission Files> gcc -o keygen BT18CSE046_AC_B_Kg.c -lgmp
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submission Files> gcc -o bob BT18CSE046_AC_B_De.c -lgmp -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submission Files> gcc -o alice BT18CSE046_AC_B_En.c -lgmp -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submission Files> ./keygen 32
PRIME q:4147312067
Done!
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submission Files>
```

- Generates both public keys and secret keys(PK.txt and SK.txt resp.)
- Run ./bob, wait for Socket created
- Run ./alice <message>

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submissio
n Files> ./alice 2345666
Conn established!
C1:3700512500, C2:2404852276
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submissio
n Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submissio
n Files> ./bob
Socket created
Connection made
Alice sent(encrpyted): 3700512500 2404852276
Decrypted:2345666
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q2\Submissio
n Files>
```

3. Entity Authentication

- **B:** Fiat-Shamir Protocol (5m)
  - **Files Provided:** BT18CSE046_EA_B_Kg.c, BT18CSE046_EA_B_B.c(Bob), BT18CSE046_EA_B_A.c(Alice)
  - **Compilation Instructions:**
    - Keygen: gcc -o keygen BT18CSE046_EA_B_Kg.c -lgmp
    - Entity with secret(Alice): gcc -o alice BT18CSE046_EA_B_A.c -lgmp -lws2_32
    - Entity who wants to verify the authenticity(Bob): gcc -o bob BT18CSE046_EA_B_B.c -lgmp -lws2_32
  - **Execution Steps:**
    - Generate keys using : ./keygen.exe <key_size>
      - Similar to elgamal
    - Run bob in listening mode: ./bob
    - Run alice: ./alice
      - We run the fiat-shamir protocol for 3 times (in a loop, with different witness every time by Alice)
  - **Sample Run:**
    - Compilation and keygen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                                                    powershell

PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submission Files> gcc -o keygen BT18CSE046_EA_B_Kg.c -lgmp
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submission Files> gcc -o alice BT18CSE046_EA_B_A.c -lgmp -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submission Files> gcc -o bob BT18CSE046_EA_B_B.c -lgmp -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submission Files> ./keygen 20
Primes p:53629, q:1015433
Done!
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submission Files>
```

    - Run ./bob, in listening mode

- Run ./alice

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submissio
n Files> ./alice
Conn established!
Round 1
Generated witness x:38278690563
Successfully sent witness x to Bob
Bob sent challenge:1
Computed response y:47445009331
Successfully sent response y to Bob
Round 2
Generated witness x:40399781127
Successfully sent witness x to Bob
Bob sent challenge:1
Computed response y:41020738229
Successfully sent response y to Bob
Round 3
Generated witness x:39998967361
Successfully sent witness x to Bob
Bob sent challenge:1
Computed response y:26888713767
Successfully sent response y to Bob
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submissio
n Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submissio
n Files> ./bob
Socket created
Connection made
Round 1
Alice sent witness x: 38278690563
Bob has choosen challenge: 1
Successfully sent challenge c to Alice
Alice sent response:47445009331
Now verifying response...
Value of y^2 mod n:45426242025
Value of x*V^c mod n:45426242025
Response verified, and is correct!
Round 2
Alice sent witness x: 40399781127
Bob has choosen challenge: 1
Successfully sent challenge c to Alice
Alice sent response:41020738229
Now verifying response...
Value of y^2 mod n:13122523237
Value of x*V^c mod n:13122523237
Response verified, and is correct!
Round 3
Alice sent witness x: 39998967361
Bob has choosen challenge: 1
Successfully sent challenge c to Alice
Alice sent response:26888713767
Now verifying response...
Value of y^2 mod n:43679953811
Value of x*V^c mod n:43679953811
Response verified, and is correct!
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q3\Submissio
n Files>
```

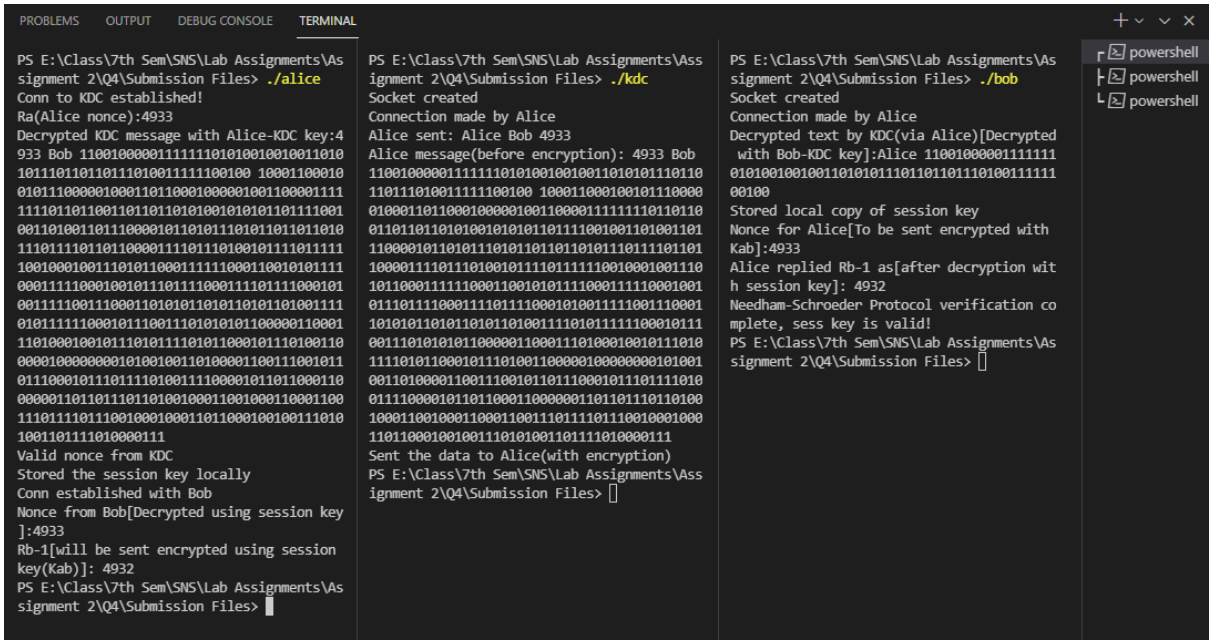
- As we can see in the above ss, all three rounds were completed successfully

4. Key Management

- **B:** Needham-Schroeder Protocol (5m)
  - **Files Provided:** BT18CSE046_KM_B_Kg.c, BT18CSE046_KM_B_Kdc.c, BT18CSE046_KM_B_A.c, BT18CSE046_KM_B_B.c
  - **Compilation Instructions:**
    - Keygen(for Alice-KDC and Bob-KDC keys): gcc -o keygen BT18CSE046_KM_B_Kg.c
    - KDC: gcc -o kdc BT18CSE046_KM_B_Kdc.c -lws2_32
    - Alice: gcc -o alice BT18CSE046_KM_B_A.c -lws2_32
    - Bob: gcc -o bob BT18CSE046_KM_B_B.c -lws2_32
  - **Execution Steps:**
    - Run keygen(generates des keys): ./keygen
    - Run bob in listening mode: ./bob
    - Run kdc in listening mode: ./kdc
    - Run alice to make session request: ./alice
  - **Sample Run:**
    - Compilation and keygen:

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q4\Submission Files> gcc -o  keygen BT18CSE046_KM_B_Kg.c
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q4\Submission Files> gcc -o kdc BT18CSE046_KM_B_Kdc.c -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q4\Submission Files> gcc -o alice BT18CSE046_KM_B_A.c -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q4\Submission Files> gcc -o bob BT18CSE046_KM_B_B.c -lws2_32
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q4\Submission Files> ./keygen
Generating Alice-KDC and Bob-KDC keys...
Done!
PS E:\Class\7th Sem\SNS\Lab Assignments\Assignment 2\Q4\Submission Files>
```

    - Run Bob on 1 terminal
    - Run kdc on another terminal, wait for Socket created
    - Run alice



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\Class\7th Sem\SNS\Lab Assignments\As
signment 2\Q4\Submission Files> ./alice
Conn to KDC established!
Ra(Alice nonce):4933
Decrypted KDC message with Alice-KDC key:4
933 Bob 110010000011111110101001001001101
0101101101101110100111111100100 10001100010
0101110000010001101100010000010011000011111
1111011011001101101101010010101010110111100 1
0011010011011100001011010111010110110110 1
1110111101101100001111011101001011110111111
1001000100111101011000111111000110010101111
0001111100010010111011110001111011110001 01
0011111001110001101010110101101010110100111 1
0101111110001011100111010101011000001100 01
1101000100101110101110101100010111010011 0
0000100000000010100100110100001100111001011
0111000101110111101001111000010110110001 10
0000011011011101101001000110010001100011 00
1110111101110010001000110100010010011101 0
1001101111010000111
Valid nonce from KDC
Stored the session key locally
Conn established with Bob
Nonce from Bob[Decrypted using session key
]:4933
Rb-1[will be sent encrypted using session
key(Kab)]: 4932
PS E:\Class\7th Sem\SNS\Lab Assignments\As
signment 2\Q4\Submission Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\Ass
ignment 2\Q4\Submission Files> ./kdc
Socket created
Connection made by Alice
Alice sent: Alice Bob 4933
Alice message(before encryption): 4933 Bob
110010000011111110101001001001101010111011 0
1101110100111111100100 10001100010010111000 0
0100011011000100000100110000111111110110110
0110110110101001010101101111001001101001101
1100001011010111010110110110101110111101101
1000011110111010010111101111110010001001110
1011000111111000110010101111000111110001001
0111011110001110111110001010011111001110001
1010101101011010110100111101011111100010111
0011101010101100000110001110100010010111010
1111010110001011101001100000100000000101001
0011010000110011100101101110001011101111010
0111100001011011000110000001101101110110100
1000110010001100011001110111011100100001000
1101100010010011101010011011110100001 11
Sent the data to Alice(with encryption)
PS E:\Class\7th Sem\SNS\Lab Assignments\Ass
ignment 2\Q4\Submission Files>
```

```
PS E:\Class\7th Sem\SNS\Lab Assignments\As
signment 2\Q4\Submission Files> ./bob
Socket created
Connection made by Alice
Decrypted text by KDC(via Alice)[Decrypted
 with Bob-KDC key]:Alice 11001000001111111
0101001001001101010111011011011001111111
00100
Stored local copy of session key
Nonce for Alice[To be sent encrypted with
Kab]:4933
Alice replied Rb-1 as[after decryption wit
h session key]: 4932
Needham-Schroeder Protocol verification co
mplete, sess key is valid!
PS E:\Class\7th Sem\SNS\Lab Assignments\As
signment 2\Q4\Submission Files>
```

powershell
powershell
powershell

- Alice stores her key in a file called Kab_Alice_copy.txt
- Bob stores his copy of the session key(generated by kdc, sent by alice, encrypted by kdc using kdc-bob key) as Kab_Bob_copy.txt
- In the above ss, it is apparent that the protocol has been successfully executed