**Eastern Mediterranean University**

**Computer Engineering Department**

**CMPE-455 Security of Computer Systems and Networks**

**Midterm Exam**

**Three A4 sheets of handwritten paper may be used for your help. Photocopies, printouts, etc. are not allowed! Calculators are allowed, other electronic devices are not allowed**

**Duration: 110 Minutes April 13, 2019**

**Std Id\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Std Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Alexander Chefranov**

**Totally 7 questions, 11 pages, 100 points**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **7.** | **Total** |
| **10** | **15** | **15** | **15** | **15** | **15** | **15** | **100** |
|  |  |  |  |  |  |  |  |

**Q1.** **(10 points).** Consider a system with the users John, Anne, and Hasan, and objects O1, O2, O3, O4, O5,O6. John can write O1 and owns O2, Anne can read O4 and O5, Hasan can read, write, and execute O1 and read O6, Anne is an owner of O~~2~~4 and O3. Construct a Capability List to keep the privileges described. Give necessary explanations.

# Q2. (15 points). Encrypt and decrypt back the following English language message “Computer Security” using substitution cipher with a key phrase: “Wikileaks founder Julian Assange arrested and carried out of Ecuador embassy”. Alphabet consists of 26 English letters and a blank symbol. Fill in the substitution table below. Give necessary explanations.

# Substitution table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | A | B | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |  |
| 2 | w | i | k | l | E | a | s |  | f | o | u | n | d | r | j | g | t | c | m | b | y | h | p | q | v | x | z |

Encryption: substitute symbols of the Row 1 by symbols from the Row 2 from the Substitution table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Plaintext | c | o | m | P | u | t | e | r |  | s | e | c | u | r | i | t | y |
| Ciphertext | k | j | d | G | y | b | e | c | z | m | e | k | y | c | f | b | v |

Decryption: substitute symbols of the Row 2 by symbols from the Row 1 from the Substitution table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ciphertext | k | j | d | G | y | b | e | c | z | m | e | k | y | c | f | b | v |
| Plaintext | c | o | m | P | u | t | e | r |  | s | e | c | u | r | i | t | y |

**Q3.** **(15 points).** DES round key generation procedure is described in the text below and Figure 3.8 from the Lecture notes:

**“KEY GENERATION**

Input key has 64 bits. But each 8th bit is not used: bits 8,16,24,32,40,48,56,64 are not further used. The 56-bit key is first subjected to permutation Permuted Choice 1:

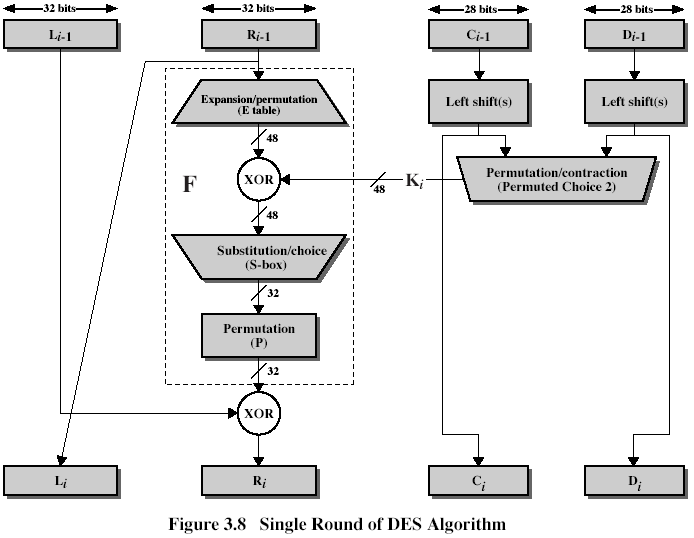
|  |
| --- |
| Permuted Choice 1 (PC-1) |
| 57 49 41 33 25 17 9  1 58 50 42 34 26 18  10 2 59 51 43 35 27  19 11 3 60 52 44 36 |
| 63 55 47 39 31 23 15  7 62 54 46 38 30 22  14 6 61 53 45 37 29  21 13 5 28 20 12 4 |

The resulting 56-bit key is then treated as two 28-bit quantities, labeled C0 and D0. At each round, Ci-1 and Di-1 are separately subjected to a circular left shift, or rotation, of 1 or 2 bits as governed by the following:

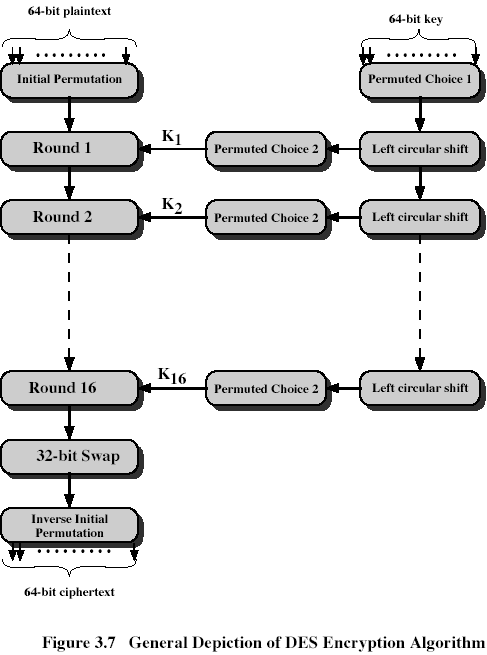
|  |
| --- |
| Schedule of Left Shifts |
| Round number 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16  Bits rotated 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1 |

These shifted values serve as input to the next round. They also serve as input to Permuted Choice 2, which produces a 48-bit output that serves as input to the function.

|  |
| --- |
| Permuted Choice 2 (PC-2) |
| 14 17 11 24 1 5 3 28  15 6 21 10 23 19 12 4  26 8 16 7 27 20 13 2  41 52 31 37 47 55 30 40  51 45 33 48 44 49 39 56  34 53 46 42 50 36 29 32 |



“

The general DES schema is

A master 64-bit key, *K*, specified by a user in hexadecimal form is as follows, K=0x02ff 1921 aac0 *aac0*. Calculate the first round-key, *K1*. Give necessary explanations

Master key:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 8 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

After PC-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | | | | | | | |
| 5 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 6 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 8 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

The upper and lower parts contain 28 bits each. Round number is 1, then according to the Schedule of left shifts, each half is shifted left 1 bit. After LS-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
|  | | | | | | | |
| 5 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 7 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 8 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

After PC-2:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

Thus, K1=0xcb 54 45 28 44 da

**Q4.** **(15 points).** Write the formula defining a multiplicative inverse in Zn. Find 123-1 mod 235 using Extended Euclid’s algorithm. Show your calculations. Check correctness of your calculations using the multiplicative inverse defining formula.

Hint:

EXTENDED EUCLID(m,b)

1. (A1,A2,A3):=(1,0,m); (B1,B2,B3):=(0,1,b);
2. if B3=0 return A3=gcd(m,b); no inverse
3. if B3=1 return B3 = gcd(m,b); B2= b-1 mod m
4. Q=
5. (T1,T2,T3):=(A1-QB1, A2-QB2, A3-QB3)
6. (A1,A2,A3):= (B1,B2,B3)
7. (B1,B2,B3):= (T1,T2,T3)
8. goto 2

A=(1,0,235), B=(0,1,123)

T=A-qB=A-B=(1,-1,112)

A=(0,1,123), B=(1,-1,112)

T=A-qB=A-B=(-1,2,11)

A=(1,-1,112), B=(-1,2,11)

T=A-qB=A-10\*B=(11,-21,2)

A=(-1,2,11), B=(11,-21,2)

T=A-qB=A-5\*B=(-56,107,1)

A=(11,-19,2), B=(-56,107,1)

As far as B3=1, B2=123-1 mod 235=107

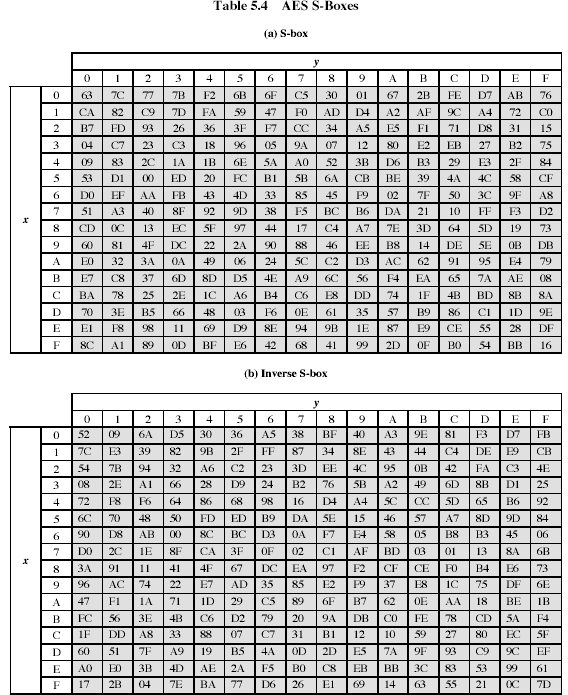
Check it by multiplication:

**Q5.** **(15 points)** Write in the placeholders below results of

- AES S-box transformation for GF(28) element {1f}: \_C0 (row 1, col F)\_\_\_\_\_\_

- inverse S-box transformation for the element {cd}: \_\_80 (row C, col D)\_\_\_\_\_\_

Hint:



**Q5. (15 points)** Consider the following materials from the Lecture Notes:

“**AES Key Expansion**

The AES key expansion algorithm takes as input a 4-word (16-byte) key and produces a linear array of 44 words (156 bytes). The following pseudo code describes the expansion:

KeyExpansion(byte key[16], word w[44]){

Word temp;

For(i=0;i<4;i++) w[i]=(key[4\*i], key[4\*i+1], key[4\*i+2], key[4\*i+3]);

For(i=4;i<44;i++){

Temp=w[i-1];

If(I mod 4 = 0) temp = SubWord(RotWord(temp)) XOR Rcon[i/4];

W[i]=w[i-4] XOR temp;

}

}

The key is copied into the 1st four words of the expanded key. The remainder of the expanded key is filled in four words at a time. Each added word w[i] depends on the immediately preceding word, w[i-1], and the word four positions back, w[i-4]. In three out of four cases, a simple XOR is used. For a word whose position in the array w is a multiple of 4, a more complex function is used. Figure 5.6 illustrates the generation of the 1st eight words of the expanded key, using the symbol g to represent the complex function. The function g consists of the following subfunctions:

1. RotWord performs a 1-byte circular left shift on a word. This means that an input word [b0, b1, b2, b3] is transformed into [b1, b2, b3, b0].
2. SubWord performs a byte substitution on each byte of its input word, using the S-box (Table 5.4a)
3. The result of steps 1 and 2 is XORed with a round constant, Rcon[j]

The round constant is a word in which the three rightmost bytes are always 0. Thus the effect of an XOR of a word with Rcon is to only perform an XOR on the leftmost byte of the word. The round constant is

different for each round and is defined as Rcon[j]=(RC[j],0,0,0), with RC[1]=1, RC[j]=2RC[j-1] and with multiplication defined over the field GF(28). The values of RC[j] in hexadecimal are

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| J | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| RC[j] | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | 1b | 36 |

For example, suppose that the round key for round 8 is

EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F

Then the 1st four bytes (1st column) of the round key for round 9 are calculated as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I(decimal) | Temp | After RotWord | After SubWord | Rcon(9) | After XOR  With Rcon | W[i-4] | W[i]=temp XOR w[i-4] |
| 36 | 7f8d292f | 8d292f7f | 5da515d2 | 1b000000 | 46a515d2 | Ead27321 | Ac7766f3 |

“

Suppose that the master key is

7F 8D 29 2F B5 8D BA D2 31 2B F5 60 EA D2 73 21

What is the round key K1 for the 1st round (w[4..7])? Show your calculations by filling the table below

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I(decimal) | Temp | After RotWord | After SubWord | Rcon(1) | After XOR  With Rcon | W[i-4] | W[i]=temp XOR w[i-4] |
| 4 | EA D2 73 21 | D2 73 21 EA | B5 8F FD 87 | 01000000 | B4 8F FD 87 | 7F 8D 29 2F | CB 02 D4 A8 |
| 5 | CD 02 D4 A8 |  |  |  |  | B5 8D BA D2 | 78 8F 6E 6A |
| 6 | 78 8F 6E 6A |  |  |  |  | 31 2B F5 60 | 49 A4 9B 0A |
| 7 | 49 A4 9B 0A |  |  |  |  | EA D2 73 21 | A3 24 88 2A |

B4 8F FD 87+7F 8D 29 2F=

1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 +

0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 0 0 1 0 1 1 1 1 =

1 1 0 0 1 0 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 0 0 0 = CD 02 D4 A8

CD 02 D4 A8 + B5 8D BA D2 =

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |  | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |  | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |  | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |  | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |  | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |  | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |

=78 8F 6E 6A

78 8F 6E 6A + 31 2B F5 60 =

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |  | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |  | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |  | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |  | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |  | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |  | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

=49 A4 9B 0A

49 A4 9B 0A + EA D2 73 21 =

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |  | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |  | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |  | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |  | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

=A3 24 88 2A

Thus K1= 0x CD 02 D4 A8 78 8F 6E 6A 49 A4 9B 0A A3 24 88 2A

**Q6. (15 points)** Let N=77. RSA encrypt and decrypt back M=2. Explain how you define RSA parameters and keys. Show your calculations. Give necessary explanations.

Hint:

“RSA (Rivest-Shamir-Adelman, 1978) algorithm is an asymmetric encryption algorithm. To design an encryption/decryption key pair, two large prime numbers, p and q, , are selected, and an integer, d, is chosen that is relatively prime to (p-1)(q-1) (d and (p-1)(q-1) have no common factors other than 1). Finally, an integer e is computed such that



One key is (e,N), and the other is (d,N), where N=p\*q, and is referred to as the modulus.

For example, we might select p=7, and q=13. Then N=91, and (p-1)(q-1)=72. We can choose d=5 (which is relatively prime to 72) and e=29, because e\*d=145 and



Then, one key is K1=(29,91) and the other is K2=(5,91). The message to be encrypted is broken into blocks such that each block, M, can be treated as an integer between 0 and (N-1). To encrypt M into the ciphertext block, B, we perform



To decrypt B, we perform



The protocol works correctly because



More details about RSA algorithm can be found in the textbook by William Stallings, Cryptography and Network Security.

Returning to the example, assume M=2.

Then, to encrypt M, we compute



Thus, B=32. To decrypt B, we compute



which is the plaintext message M.”

N=77=7\*11 => p=7, q=11

Fi(N)=(p-1)\*(q-1)=6\*10=60

E=7: gcd(E,fi(N))=gcd(60,7)=gcd(7,4)=gcd(4,3)=gcd(3,1)=gcd(1,0)=1

D=E-1 mod 60 = 7-1 mod 60 =

A=(1,0,60), B=(0,1,7), q= 8

T=A-qB=(1,-8,4)

A=(0,1,7), B=(1,-8,4), q=1

T=A-qB=(-1,9,3)

A=(1,-8,4), B=(-1,9,3), q=1

T=A-qB=(2,-17,1), Hence,

Actually, 7\*43=301=5\*60+1 =1 mod 60.

Encrypt:

Decrypt:

43=32+8+2+1, hence,

Thus, , and the plaintext is disclosed back.

T=A-qB=(1,-8,4)

A=(0,1,7), B=(1,-8,4), q=1

**Q7. (15 points)** Consider Diffie-Hellman key exchange protocol below from the Lecture notes:

Let q=11. Build an example showing how two parties, A and B, get the same key, K. Show your calculations. Give necessary explanations.

Let

Then, the key obtained by A is and the key obtained by B is