**Eastern Mediterranean University**

**Computer Engineering Department**

**CMSE-353 Security of Software Systems**

**Final Exam**

**Ten A4 sheets of handwritten paper may be used for your help. Handwritten copies, photocopies, printouts, books, telephones, etc. are not allowed! Calculators are allowed, other electronic devices are not allowed.**

**Duration: 180 Minutes January 3, 2019**

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

**Instructor Alexander Chefranov**

**Totally 10 questions, 100 points, 18 pages**

**Questions Q1-Q5 (33 points) cover before- and Q6-Q10 (67 points) after-MT material**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question**  **(max point)** | **Q1**  **(6)** | **Q2**  **(6)** | **Q3**  **(7)** | **Q4**  **(7)** | **Q5**  **(7)** | **Q6**  **(13)** | **Q7**  **(13)** | **Q8**  **(13)** | **Q9**  **(14)** | **Q10**  **(14)** | **Total**  **(100)** |
| **Point** |  |  |  |  |  |  |  |  |  |  |  |

**Q1. (6 points)** What is an insider attack? Enlist four counter-measures against insider attack

An insider attack is conducted by a legal user willing to go beyond his allowed permissions. Counter-measures are:

1. Restrict staff permissions to the minimal required by his/her duties
2. Critical actions such as backups shall be in responsibility of two people
3. Physically secure critical systems
4. Monitor employees’ behavior

**Q2 (6 points)**. For the Hill cipher with the key matrix,

=

|  |  |  |  |
| --- | --- | --- | --- |
| K= | 17 | 17 | 5 |
| 21 | 18 | 21 |
| 2 | 2 | 19 |

Assuming only English alphabet letters, encrypt plaintext, p=”key”. Show your calculations. Give explanations for your answer.

Hint:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

P=”key” = (10, 4, 24). Operations will be done modulo n=26 because the alphabet has 26 symbols

C=K\*P

C1=k11\*p1+k12\*p2+k13\*p3 mod n=17\*10+17\*4+5\*24 mod 26 = 170+68+120 mod 26 = 170 mod 26 + 68 mod 26 +120 mod 26 = 14+16+16=46 mod 26 = 20

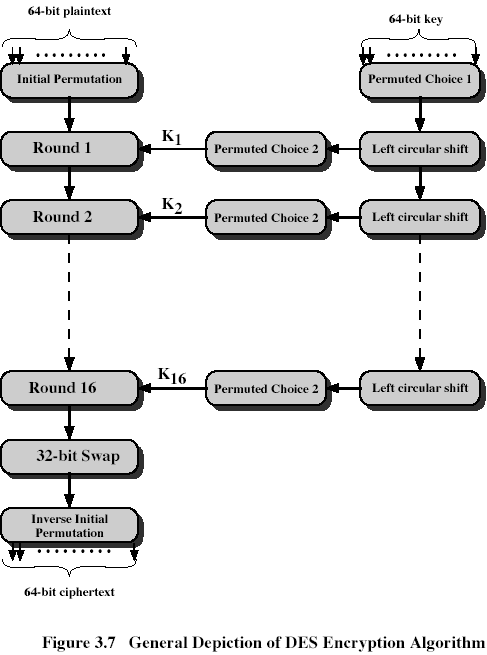
C2=k21\*p1+k22\*p2+k23\*p3 mod n=21\*10+18\*4+21\*24 mod 26 = 210+72+504 mod 26 = 210 mod 26 + 72 mod 26 +504 mod 26 = 2+20+10=32 mod 26 = 6

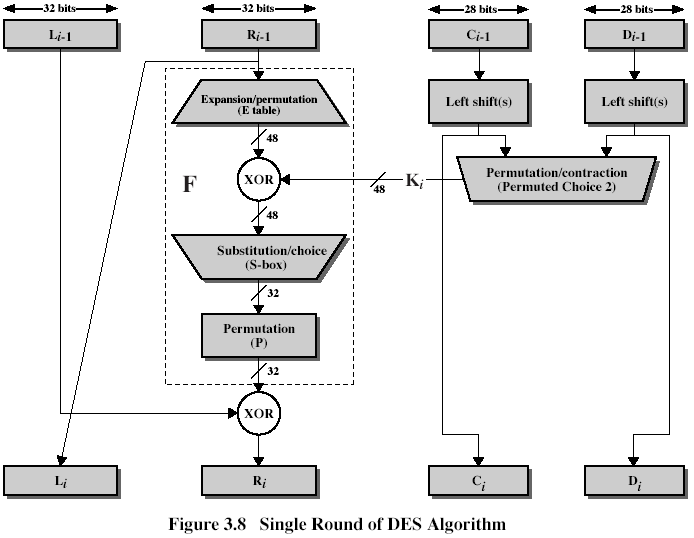
C3=k31\*p1+k32\*p2+k33\*p3 mod n=2\*10+2\*4+19\*24 mod 26 = 20+8+456 mod 26 = 28 mod 26 + 456 mod 26 = 2+14=16

Thus, C=(20, 6, 16) = “ugq”

**Q3 (7 points).** What S-boxes in the next round of DES are affected by S-box S3 of the previous round? Give explanations for your answer.

Hints:





|  |  |  |
| --- | --- | --- |
| Expansion/Permutation (E table) | | |
| 32 | 1 2 3 4 | 5 |
| 4 | 5 6 7 8 | 9 |
| 8 | 9 10 11 12 | 13 |
| 12 | 13 14 15 16 | 17 |
| 16 | 17 18 19 20 | 21 |
| 20 | 21 22 23 24 | 25 |
| 24 | 25 26 27 28 | 29 |
| 28 | 29 30 31 32 | 1 |



|  |
| --- |
| Permutation function( P ) |
| 16 7 20 21 29 12 28 17  1 15 23 26 5 18 31 10  2 8 24 14 32 27 3 9  19 13 30 6 22 11 4 25 |

S3 outputs bits 9, 10, 11, 12. According to permutation, P, they are placed in the positions 24, 16, 30, and 6, respectively. After Expansion/Permutation in the next round they are placed in the positions: 24 is in the 6th row, hence, it affects S6, and is in the 7th row, hence, it affects S7; 16 is in the 4th row, hence, it affects S4, and is in the 5th row, hence, it affects S5; 30 is in the 8th row, hence it affects S8; 6 is in the 2nd row, hence it affects S2. Thus, S3 affects six S-boxes: S6, S7, S4, S5, S8, and S2.

**Q4 (7 points).** Find . Show your calculations, explain your answer

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, 122), B=(0,1,11)

Q=A3/B3=11

T=A-QB=(A1-QB1, A2-QB2, A3-QB3)=(1, -11, 1)

A=(0,1,11), B=(1, -11, 1)

B3=1 => B2=-11=11-1mod122=111

Actually, 11\*111=1110+111=1221; 1221 =122\*10+1 = 1 mod 122

**Q5. (7 points)** What is a mix column transformation in AES? Show that the state array element in the 1st row and 2nd column is actually {40} as it is shown in the example below

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 87 | F2 | 4D | 97 |  | 47 | 40 | A3 | 4C |
| 6E | 4C | 90 | EC | => | 37 | D4 | 70 | 9F |
| 46 | E7 | 4A | C3 |  | 94 | E4 | 3A | 42 |
| A6 | 8C | D8 | 95 |  | ED | A5 | A6 | BC |

Show your calculations, give explanations.

Hint: **Mix Column Transformation**

The forward mix column transformation, called MixColumns, operates on each column individually. Each byte is mapped into a new value that is a function of all four bytes in the column. The transformation can be defined as the following matrix multiplication on State (Fig. 5.5b):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |  | S00 | S01 | S02 | S03 |  | S00’ | S01’ | S02’ | S03’ |  |
| 01 | 02 | 03 | 01 | \* | S10 | S11 | S12 | S13 | = | S10’ | S11’ | S12’ | S13’ | (5.3) |
| 01 | 01 | 02 | 03 |  | S20 | S21 | S22 | S23 |  | S20’ | S21’ | S22’ | S23’ |  |
| 03 | 01 | 01 | 02 |  | S30 | S31 | S32 | S33 |  | S30’ | S31’ | S32’ | S33’ |  |

Each element in the product matrix is the sum of products of elements of one row and one column. In this case, multiplications and additions are performed in GF(28); 

1st column of the result is obtained by:

{02){87}+{03}{6E}+{46}+{A6} = {47}

{87}+{02}{6E}+{03}{46}+{A6} = {37}

{87}+{6E}+{02}{46}+{03}{A6} = {94}

{03}{87}+{6E}+{46}+{02}{A6} = {ED}

For the 1st equation, we have {02}{87}=(0000 0010)(1000 0111)=

=

(0001 0101)={15}

{03}{6E}=(0000 0011)(0110 1110)= =

(1011 0010) = {B2}

{02){87}+{03}{6E}+{46}+{A6}={15}+{B2}+{46}+{A6}=

(0001 0101)+

(1011 0010)+

(0100 0110)+

(1010 0110)=

(0100 0111)={47}

We need proving that

{02){F2}+{03}{4C}+{E7}+{8C} = {40}

{02}{F2}=(0000 0010)(1111 0010)=

=

(1111 1111)={FF}

{03}{4C}=(0000 0011)(0100 1100)= =

(1101 0100) = {D4}

~~{02){87}+{03}{6E}+{46}+{A6}={15}+{B2}+{E7}+{8C}=~~

{02){F2}+{03}{4C}+{E7}+{8C}={FF}+{D4}+{E7}+{8C}=

(1111 1111)+

(1101 0100)+

(1110 0111)+

(1000 1100)=

(0100 0000)={40}

**Q6. (13 points)** Encrypt and decrypt back the plaintext M=2 using RSA with N=77. Define all necessary RSA parameters, give 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=p\*q=11\*7 => p=11, q=7

Fi(N)=(p-1)(q-1)=60

Let e=7, gcd(e, fi(N))=gcd(7,60)=1, hence e-1mod fi(N) exists, and it is e-1 =d=43. Actually, 7\*43=301=5\*60+1-1 mod 60

It can be found EEA:

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

Q=A3/B3=8

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

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

Q=A3/B3=1

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

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

Q=A3/B3=1

T=A-QB=(2, -17, 1) => 7-1 mod 60 =-17 mod 60=43

Encryption: c=me mod N= 27 mod 77 = 128 mod 77 = 51

Decryption: m’ = cd mod N =5143 mod 77 = 2

512 mod 77 = 60

514 mod 77 = 602 mod 77 = 58

518 mod 77 = 582 mod 77 = 53

5116 mod 77 = 532 mod 77 = 37

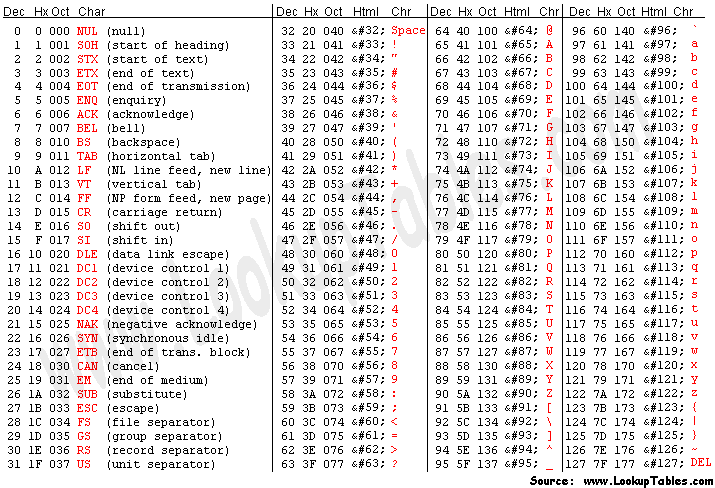
5132 mod 77 = 372 mod 77 = 60

Hence, 5143 mod 77 = 5132 \*518 \*512 \*51 mod 77 = 60\*53\*60\*51 mod 77 = (360 mod 77) \* 10\*53\*51 mod 77 = 52\*53\*51\*2\*5 mod 77 = 52\*53\*(102 mod 77) \*5 mod 77 = 52\*53\*25 \*5 mod 77= 52\*53\*(125 mod 77) mod77 = 52\*53\*48 mod 77 = (52\*2)\*53\*24 mod 77 = 104 mod 77 \* 53\*24= 27\*53\*24 mod 77 = 9\*(3\*53)\*24 mod 77 = 9\*(159 mod 77) \*24 mod 77 = 9\*5\*24 mod 77= 9\*(120 mod 77 ) mod 77 = 3\*(3\*43)= 3\*(129 mod 77) mod 77 = 3\*52 mod 77 = 156 mod 77 = 2=M

**Q7. (13 points)** Explain how Protected authentication is conducted with two functions, f1 and f2? Assume, Password=”12”, A=”EMU”, t1=”1”, t2=”2”, q1=”3”, q2=”4”, f1(STR)=x mod 256, f2(STR)=(2x+1) mod 256, where x is the sum of ASCII codes of STR. If the functions have several input strings, they are concatenated. Calculate Protected2.Show your calculations.

HINT:





Example: f1(“USER”+”12”)=f1(85+83+69+82+49+50)=418 mod 256 = 162.

Protected authentication is conducted as follows. At first, f1(A, pwd, t1, q1)= Protected1 is calculated. Then, f2(Protected1, t2, q2)=Protected2 is calculated by the client, and the client transfers to the server the following data: A, pwd, t1, q1, t2, q2, Protected2. The server, using the client’s name, A, finds in its database, the client’s password, P, and calculates f2(f1(A, P, t1, q1), t2, q2)=V. If V==Protected2 received, the client is accepted, otherwise, rejected.

Protected1=f1(“EMU”+”12”+”1”+”3”)=f1(“EMU1213”)=(69+77+85+49+50+49+51) mod 256= (146+134+150) mod 256 = (280+150) mod 256 = (24+150) mod 256 = 174

Protected2=f2(“174”+”2”+”4”)=f2(“17424”)=(2\*(49+55+52+50+52)+1) mod 256 = (2\*(104+154)+1) mod 256 = (2\*258+1) mod 256 = (2\*(258 mod 256) + 1) mod 256 = 2\*2+1 = 5.

**Q8. (13 points)** Calculate F(B,C,D) for B=11010, C=00101, D=11101. Show your calculations, give explanations

Hint:

|  |  |  |
| --- | --- | --- |
| Round | Primitive function g | g(b,c,d) |
| 1 | F(b,c,d) |  |
| 2 | G(b,c,d) |  |
| 3 | H(b,c,d) |  |
| 4 | I(b,c,d) |  |

F(b1, c1, d1)= F(1,0,1)=(1^0)v(0^1)=0v0=0

F(b2, c2, d2)= F(1,0,1)=(1^0)v(0^1)=0v0=0

F(b3, c3, d3)= F(0,1,1)=(0^1)v(1^1)=0v1=1

F(b4, c4, d4)= F(1,0,0)=(1^0)v(0^0)=0v0=0

F(b5, c5, d5)= F(0,1,1)=(0^0)v(1^1)=0v1=1

Thus, F(B,C,D)=00101

**Q9. (14 points)** Assume, that redundancy predicated for serial numbers is p(x)=13|x. Using Anonymous Protocol, generate a serial number and get a token using RSA settings and keys generated in **Q6**. Show your calculations, give explanations.

HINT:

C first generates a random number, *u*, that is relatively prime to the modulus *N* of the bank’s keys. Because u is relatively prime to *N*, it has a multiplicative inverse, , with respect to *N*, such that



To blind the serial number, *n*, C computes



and sends the result to B. Hence, the blinding function can be viewed simply as multiplication by a random number.

The signed result, *sr*, returned by B to C is



Obviously, . To recover the token, we use



The serial number *n* can be now obtained using.

Let n=26: p(n)=13|26=true. Use RSA parameters from Q6: p=11, q=7, N=77, e=7, d=43

Let u=2, gcd(u,N)=gcd(2,77)=1, u-1 mod N= 2-1 mod 77= 39. Actually, 2\*39=78=1 mod 77

Let P=e=7, R=d=43

Then, b(n)=RSAP(2)\*n mod N = (27 mod 77 \*26) mod 77 = (51\*26) mod 77 = (17\*(3\*26 mod 77) ) mod 77 = 17

Sr= RSAP(17)= 1743 mod 77 =

172 mod 77 = 58

174 mod 77 = 53

178 mod 77 = 37

1716 mod 77 = 60

1732 mod 77 = 58

1743 mod 77 =1732 \*178 \* 172 \*17mod 77 =58\*37\*58\*17 mod 77 = 67\*58\*17 mod 77 = 36\*17 mod 77 = 73

Recovered token, RSAR(26)=39\*73 mod 77 = 75 = 2643 mod 77

262 mod 77 = 60

264 mod 77 = 58

268 mod 77 = 53

2616 mod 77 = 37

2632 mod 77 = 60

2643 mod 77 = 2632 \* 268 \* 262 \*26mod 77 = 60\*53\*60\*26 mod 77 = 23\*60\*26 mod 77 = 71\*26 mod 77 = 75

**Q10. (14 points)** Calculate the first nLookupIndex value obtained in CRC32 function assuming byte=123. Show your calculations, give explanations

HINT:

**Function** CRC32

**Input:**

Data: Bytes **Array of bytes**

**Output:**

crc32: UInt32 **32-bit unsigned crc-32 value**

**Initialize crc-32 to starting value**

crc32 ← 0xffffffff

**for each** byte **in** data **do**

nLookupIndex ← (crc32 xor byte) and 0xFF;

crc32 ← (crc32 shr 8) xor CRCTable[nLookupIndex] **//CRCTable is an array of 256 32-bit constants**

123 = 7\*16 + 11 = 0x7B

nLookupIndex=(0xFFFFFFFF xor 123 ) and 0xFF = (0xFFFFFFFF xor 0x0000007B) and 0xFF =0xFFFFFF84 and 0x000000FF = 0x84 = 16\*8+4 = 152

0xFFFFFFFF xor 0x0000007B=

=

1111 1111 1111 1111 1111 1111 1111 1111

xor

0000 0000 0000 0000 0000 0000 0111 1011

=

1111 1111 1111 1111 1111 11111000 0100 = 0xFFFFFF84

0xFFFFFF84 and 0x000000FF =

=

1111 1111 1111 1111 1111 11111000 0100

and

0000 0000 0000 0000 0000 0000 1111 1111=

=

0000 0000 0000 0000 0000 0000 1000 0100= 0x00000084 = 0x84