**Problem Session CMSE-353 “Security of Software Systems” 15.01.2021**

AES, Hill cipher, RSA, Digital signatures, Certificates, SSL, Microsoft Password

1. RSA: key generation, Euclid GCD algorithm, Extended Euclidean algorithm, finding multiplicative inverse, Euler totient function, encryption/decryption, exponentiation by squaring with in-line reducing

Keys shall be relatively prime to fi(N)=(p-1)(q-1)=fi(p)\*fi(q), fi is Euler totient function; N=p\*q

Gcd(a,b)=1=>a,b are relatively prime; gcd(a,0)=a

Gcd(103, 24)=gcd(24, 103 mod 24)=gcd(24, 7)=gcd(7, 24 mod 7)=gcd(7,3)=gcd(3,7 mod 3)=gcd(3,1)=gcd(1,0)=1

Gcd(a,b)=gcd(b, a mod b = a-q\*b, q=floor(a/b))

E\*d=1 mod fi(N)

Extended Euclidean algorithm (m,b) b^(-1) mod m

A=(A1,A2,A3)=(1,0,m); B=(B1,B2,B3)=(0,1,b)

Step 1: B3=0=> A3 is gcd(A3,B3)=gcd(m,b)<>1 => inverse does not exist

B3=1=>gcd(m,b)=1=>inverse exists and b^(-1) mod m= B2; B1=m^(-1) mod b

Iteration

Q=floor(A3/B3);

T=A-q\*B=(A1-q\*B1, A2-q\*B2, A3-q\*B3 = A3 mod B3)

A=B;

B=T;

Goto Step 1;

C=M^e mod N; M=C^d mod N

Exponentiation use squaring together with reduction

M^(2k) mod N= (M^k mod N)^2 mod N

29=16+8+4+1

M^29 mod N=M^(16+8+4+1) mod N=(M^16 mod N)\*(M^8 mod N)\*(M^4 mod N)\*M

M^2 mod N =M2<N

M^4 mod N = M2^2 mod N =M4

M^8 mod N=M4^2 mod N =M8

M^16 mod N = M8^2 mod N =M16

M^29 mod N=) (M16\*M8 mod N)\*M4 mod N)\*M mod N

1. RSA uses: data hiding, key exchange, digital signature, hash function h(x) = x mod 11

If H(M=16)=5, M’: H(M’)=5=> M’=27 mod 11 =5

Modes of DES operation: CFB, OFB, Counter

B1=>C1 B2=C1 B3=>C3 ECB K

H(B1,B2,B3, K)= C1 xor C2 xor C3

1. Certificates, Certificate authority (CA), Subject, Public key, version, serial number, issuer, validity period, CA web-site, revocation list
2. Secure sockets layer and Microsoft password authentication protocols. Relation between them

Certificate Cert of the server S => client, client validates Cert, generates session key, Cipher=EPublic key(Session key), transmits Cipher to S, S decrypts Cipher with the private key, gets Session key, and communicates to client using AES with the session key: asks user\_name, pwd, etc

1. AES: Encryption/Decryption schemes, rounds, plaintext, state array plaintext 128 bits, 16 bytes, 4 words (32-bit) column major

|  |  |  |  |
| --- | --- | --- | --- |
| W[0] | W[1] | W[2] | W[3] |
| B0 | B4 | B8 | B12 |
| B1 | B5 | B9 | B13 |
| B2 | B6 | B10 | B14 |
| B3 | B7 | B11 | B15 |

Key 128 =16 bytes = 4 words=>44 words=11 round keys (each round 4 words),192, 256 bits

Words[0-3]=master key 4 words

For(i=0;i<4;i++)w[i]=keyword[i]=(byte(4\*i+j, j=0,..,3))=(4\*I,4i+1,4i+2,4i+3);

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

 Temp=w[i-1]; when i=4=>i-1=3

If(I mod 4 =0) temp=???

W[i]= w[i-4] xor temp; = w[i-1] xor w[i-4];

1. , forward and inverse state transformations, add round key, substitute byte, shift row, mix column, GF(2^8) elements, polynomials of order at most 7=8-1 with binary coefficients (Z2={0,1})

P(x)=a7\*x^7+a6\*x^6+a5\*x^5+ a4\*x^4+a3\*x^3+a2\*x^2+a1\*x+a0 =[a7,..,a0] =>byte {xy} x, y are hexadecimal digits 0,1,.,9, a=10,b=12,c=12,d=13,e=14,f=15

1. irreducible polynomial, addition/subtraction, multiplication/division, remainder after division. Key expansion procedure, initialization, recursive procedure, RotWord, SubByte, round constant
2. Hill cipher, vector-matrix multiplication, encryption/decryption, matrix inversion, determinant
3. *Matrix inversion (for Hill ciphers*)

 - ? 

 n=10

detA=45+84+96-105-48-72=225-225=0

It means that inverse of the matrix does not exist

1. *Matrix inversion*

-? n=10



 (1)

where - is a determinant of sub matrix of A, obtained by deletion of i-th row and j-th column, det(A) – determinant of A. Taking into account that we work with integers on modulo n, we rewrite (1):

 (2)

det(A) =40+84+96-105-64-48=220-217=3

gcd(det(A),n)=1

From (2):



















Thus, we get



and

