**Problem Session CMSE-353 “Security of Software Systems” 25.12.2023**

[AES Part 2](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE353/AES%20CIPHER.Part2.doc) (p. 2-4 starting by Shift row transformation till equation (5.5) including;  p. 6-7 starting from Add Round Key transformation), [RSA Algorithm](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE353/CMSE353%20Fall2023/RSA%20algorithm.docx), [Digital Signature](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE353/Digital%20Signatures.docx), [Certificates](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE353/Certificates.docx), [SHA-512](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE512/Spring%202022/SHA512%2028032022.docx), [Authentication procedures, One-Time password](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE353/CMSE%20353%20Fall2021/Authentication%20Procedures%3D1.docx) (p. 1-6, MD5 is not included), and [Ch. 5. Network security](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE353/CMSE%20353%20Fall2021/Ch%205%20Network%20Security%2019122017.docx) (p. 1-10, 5.3 Network layer and the next are not included)

1. AES. Shift rows transformation.

Shift rows transformation shifts left circularly (rotates left) i-th row elements (bytes) i positions left. Inverse shift rows transformation shifts right circularly (rotates right) i-th row elements (bytes) i positions right. Rows are numbered from 0 to 3. Hence, the top row number 0 is not touched.

1. 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).

The following is the example of MixColumns;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

“

Be able calculating any element of the state array in the right above given the left array for the Forward Mix Column Transformation, or, vice versa, for the Inverse Mix Column Transformation.

Resulting elements of the product of the constant mix column matrix, MC, and the state array, s, are calculated as

$s\_{ij}^{'}=\sum\_{k=0}^{3}MC\_{ik}∙s\_{kj}, i=0..3, j=0..3$,

where multiplications and additions are made in $GF(2^{8})$ with the irreducible polynomial $m\left(x\right)=x^{8}+x^{4}+x^{3}+x+1$

1. AES. Consider

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;

 }

}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 XORWith Rcon | W[i-4] | W[i]=temp XOR w[i-4] |
| 36 | 7f8d292f | 8d292f7f | 5da515d2 | 1b000000 | 46a515d2 | Ead27321 | Ac7766f3 |

What are the second four bytes of round 9?

According to the code, W[i]=w[i-4] XOR temp where i=37, temp=w[i-1], thus, w[37]=w[33] xor w[36] = B58DBAD2 + Ac7766f3

=1011 0101 1000 1101 1011 1010 1101 0010

+1010 1100 0111 0111 0110 0110 1111 0011=

=0001 1001 1111 1010 1101 1100 0010 0001=19fadc21

1. RSA: For p and q from [15,20], define RSA pair of keys, encrypt and decrypt

 M=11.

P=17, q=19, n=p\*q=17\*19=323, fi(n)=16\*18=288, e=5, d=e^(-1) mod 288

Use Extended Euclid algorithm:

A=(1,0,288) B=(0,1,5)

Q=floor(A3/B3)=floor(288/5)=57

T=A-q\*B=(1-57\*0, 0-57\*1, 288-57\*5)=(1,-57,3)

A=B=(0,1,5), B=T=(1,-57,3)

Q=floor(5/3)=1

T=A-q\*B=(0-1,1-(-57),5-3)=(-1,58,2)

A=B=(1,-57,3), B=(-1,58,2)

Q=floor(A3/B3)=floor(3/2)=1

T=A-q\*b=(1-(-1),-57-58,3-2)=(2,-115,1)

Since new B3=T3=1, 5^(-1) mod 288=B2=T2=-115 mod 288 = 173

Check it: 173\*5=865=3\*288+1 mod 288 = 1, hence, it is correct, and d=173

Encrypt:

C=11^173 mod 323=11^(128+32+8+4+1) mod 323= (11^128 mod 323 \* 11^32 mod 323 \* 11^8 mod 323 \*11^4 mod 323 \* 11) mod 323

Use squaring and modulo reduction:

11^2 mod 323 =121

11^4 mod 323 =121^2 mod 323 = 106

11^8 mod 323 =106^2 mod 323 = 254

11^16 mod 323 =254^2 mod 323 = 239

11^32 mod 323 =239^2 mod 323 = 273

11^64 mod 323 =273^2 mod 323 = 239

11^128 mod 323 =239^2 mod 323 = 273

C= 11^173 mod 323 = 273\*273\*254\*106\*11 mod 323 = 239\*254\*106\*11 mod 323 = 305\*106\*11 mod 323 = 30\*11 mod 323 = 330 mod 323 = 7

M’=C^5 mod 323 = 7^5 mod 323

7^2 mod 323 = 49

7^4 mod 323 = 49^2 mod 323 = 140

7^5 mod 323 = (7^4 mod 323\*7) mod 323 = 140\*7 mod 323 = 980 mod 323 = 3\*323+11 mod 323 = 11 =M, hence, encryption and decryption are correct.

1. RSA: What is public key? What is private key? What for private key is used? What for public key is used?

Public key is published, private key is kept secretly/ Private key is used for decryption of the secret encrypted with the public key, and for encryption for digital signature. Public key is used for data encryption for secrecy, and for digital signature verification.

1. Digital signature: Using result of Task 2, and hash(x)= x mod 5, digitally sign M=28.

Digiatl signature DS= E(Private\_key, h(message)). H(message)= 28 mod 5 = 3

Private key= 5, hence DS=3^5 mod 323 = 243 mod 323 = 243.

1. Digital signature: How in SSL protocol a client authenticates a server? How a server authenticates a client?

A client authenticates a server when getting request from it encrypted with the session key generated by the client and sent to the server encrypted with the server’s public key known from its certificate. The server authenticates the client when getting from it correct user name and password encrypted with the correct session key.

1. Digital signature: How a server authenticates a client in Microsoft Passport protocol?

A server authenticates a client using service of the authentication server.

1. Certificates: What for certificates are used? Who issues them? How certificates are protected from forge? How certificate is validated? What three checks are necessary to validate a certificate?

Certificates are used to certify public key of the subject. Certificate is issued by a certificate authority (CA). Certificates are protected from forge by the digital signature of the CA.

1. Authentication procedures: What is simple unprotected authentication procedure?

Simple unprotected authentication procedure is such using transfer of a user name and password in clear, not protected.

1. Authentication procedures: How in simple unprotected authentication client is authenticated to the server?

A client is authenticated by checking provided password versus the one kept in the server’s database of the users and passwords.

1. Authentication procedures: How in simple protected authentication client is authenticated to the server?

A client is authenticated by checking provided hash of the password together wıth user name, timestamp, and nonce versus the one recalculated by the server using provided user name, timestamp, and nonce, and the password from the server’s database of users and passwords.

1. Authentication procedures: Who is authenticated in X.509 one-way authentication? In two-way authentication? In three-way authentication?

In one-way, a client is authenticated by a server, in two-way, also a server is authenticated by the client. In three-way authentication, additionally, the client also confirms authentication of the server.

1. Authentication procedures: What is the flaw of the three-way authentication procedure? What is the reason for it? How can it be fixed?
2. Authentication procedures: How client is authenticated to server in Lamport’s One-Time password? Why re-initialization is necessary after N authentications?

Authentication procedures: How the small number attack is conducted on OTP?

A client is authenticated by the server by providing a value hash of which is equal to the current password kept in the server’s database. Re-initialization is necessary, since h^N(p0) is used as the initial password, and after each authentication it is replaced the values such that hash of it was equal to the password. Eventually, the password will become h(p0), and then further authentication is not possible.

1. SHA-512: Messages of what sizes can be used as input to SHA-512? How original message length is saved in SHA-512 padding process? What is the length of a message after appending? How appending is conducted? Why in the result 0’s and 1’s padding, the message length shall be congruent to 896 mod 1024 bits? What happens if the original message length is divisible by 1024?

Any size messages can be used as input. Original message length is saved in the last 128 bits of the last block of the padded message. Appending is conducted by extending the original message by the sequence 1000.., such that the length is congruent to 896 mod 1024: $length≡896 mod 1024$. It shall be congruent to 896 to leave the last 128 bits to keep the original message size. If the original message size is divisible by 1024, still it is padded.

1. How a message is processed in SHA-512? What two inputs are used by each block in the processing chain? What is the resulting hash? What is the number of bits used in each input of SHA-512 block? How 512-bit input is represented? How IV is initialized?

A message is processed using a chain of elementary hash functions blocks. Each block in the chain gets current block and output of the previous in the chain block. The resulting hash is the output of the last block. One input of a hash block is gets 1024 bits of the current message block, and the second input is 512-bit output of the previous block output. IV is initialized by the first 64 bits of the fractional parts of the square roots of the first eight prime numbers: 2, 3, 5, 7. 11, 13, and 17, and 19. How decimal fractional part can be converted to hexadecimal system?

1. How many rounds has the SHA-512 block? What are the inputs/outputs of the rounds? How SHA-512 output is obtained? What words, constants, and logical functions are used in the rounds? How constants K are defined? What transformations are made in each round?

Each hash block has 80 rounds. Inputs of the round is outputs of the previous round which are chained. Input of the first round is output of the previous block. Output of the last round is added with the input to the block yielding the block output. Constants K are obtained from the first 64 bits of the fractional parts of the cube roots of the first 80 prime numbers.

1. Network security: five layer Internet structure, IP addresses, IPv4, IPv6, hosts, routers, hops, packets, headers, footers, payload

Five layers: Application, transport, network, link, and physical layer. IPv4 address is 32-bit, and IPv6 address is 128-bit. Hosts are computer connected via Internet. Routers provide connection of the hosts. Hops are intermediate routers via which hosts are connected. Packets are data structure used in network communications. Headers are data structures used at beginning of the packets. Footers are data structures used at the end of the packets. Payload is content of a packet located between a header and footer.

1. Network security: Ethernet, common bus, star topology, hubs, switches, MAC addresses, MAC address structure, local MAC addresses administering

Ethernet is the link layer standard. It uses common bus architecture. Common bus architecture can be represented by star topology with a hub in its center. Hub broadcasts messages from the hosts connected to it to other hosts. Switch is similar to hub but transfers messages to a particular host using MAC addresses. MAC address is a 48-bit number, first 3 bytes is organizationally unique identifier, and the 2nd 3 bytes is Network interface controller specific. Local MAC administering or globally unique identifier is defined by the 2nd least significant bit of the most significant 1st byte.

1. Network security: Ethernet frame structure, CRC32, ARP Protocol, ARP request, ARP reply, ARP cache, ARP Spoofing, ARP cache poisoning

Ethernet frame is a packet of Ethernet, its structure has a header, payload, and footer. The footer has CRC32 has function of the packet to check its authenticity. ARP protocol is address resolution protocol allowing associating respective IP and MAC addresses of the host. ARP protocol has ARP request and ARP reply packets. ARP request asks hosts for MAC address corresponding to the IP address to be resolved. ARP reply is responded to the requestor by a host with the requested IP address and has respective MAC address. ARP spoofing is an attack on ARP protocol where incorrect MAC address is transferred. ARP cache poisoning means replacing of the correct MAC address by an incorrect one in an internal table of the host.