Unit 3 PDF

Cryptography & Network Security (Jawaharlal Nehru Technological University, Hyderabad)

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Principles of Public-key Cryptosystem

Unit No:	III
Lecture No:	L27
Link to Session Planner (SP): S.No:...of SP	39
Book Reference:	T1,169
Date Conducted:	31/8/18
Page No:	41

Principle of Public-key crypto system -
The Concept of Public key evolved from an attempt to attack two most difficult ~~pb~~ problems associated with symmetric encryption
* the first problem is Key distribution
* Authentication

Diffie and Hellman achieved an astounding breakthrough in 1976. by comming up with a method that address both problems and that was radically different from all previous approaches to Cryptography.

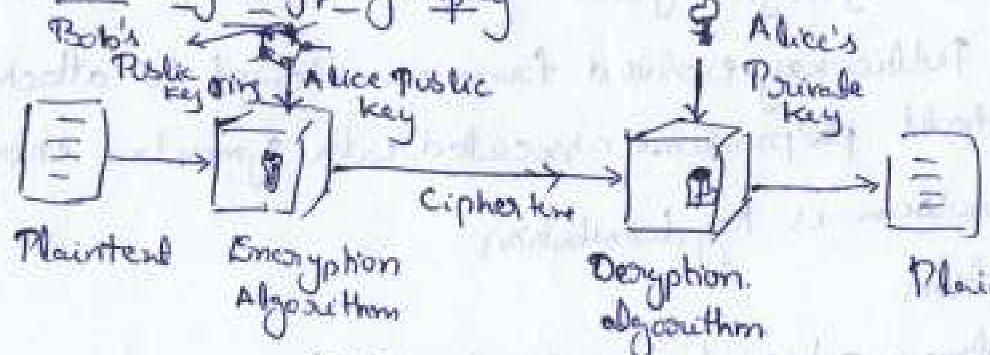
Public key cryptosystem :- Asymmetric algorithms rely on one key for Encryption and a diffrent ~~key~~ but related key for decryption.
- These algorithm have following important characteristic:-
  * → It is computationally infeasible to determine the decryption key given only knowledge of the Cryptographic algorithm and the encryption key
  * → Either of the two related keys Can be used for encryption, with the other used for decryption.
  * → It is computationally easy to en/decrypt messages when the relevent key are known.

A public-key encryption scheme has six ingredients
* Plain text - This is readable message or data input to algorithm
* Encryption Algorithm - The ~~a~~Encryption algorithm performs tranformations on the input Plain text
* Public & private key - This is a pair keys that have been selected so that one is used for Encryption other for decryption.

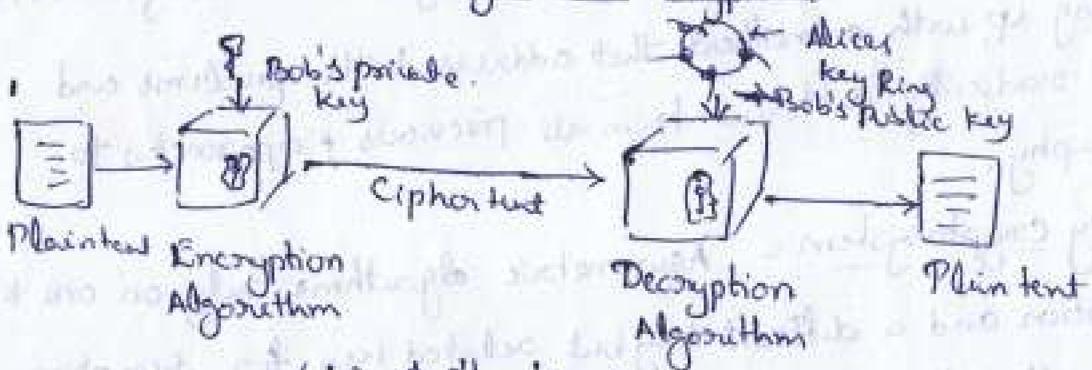
Ciphertext :- This is the scrambled message produced as output.

- Decryption algorithm :- This algorithm accepts the ciphertext and matching key and produces the original plain text.

→ Public key cryptography



(a) Encryption & decryption



(b) Authentication

The essential step's are following

(1) Each user generates a pair of key's to be used for the encryption & decryption of message.

(2) Each user places one of the two key's in a public register or other accessible file. This is the public key. The companion key is kept private.

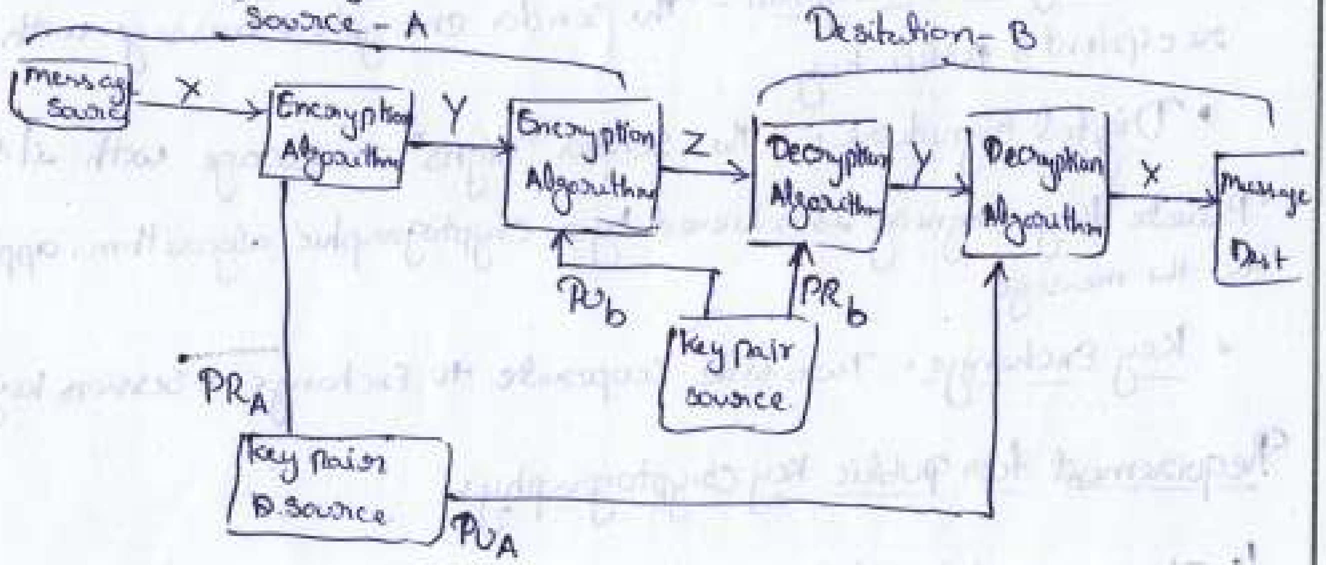
(3) If Bob wishes to send confidential message to Alice, Bob encrypts the message using Alice's public key.

(4) When Alice receives the message, decrypts it using private key. No other recipient can decrypt the message.

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Principles of public key cryptosystem:

Unit No: III
Lecture No: L27
Link to Session Planner (SP): S.No. 32 of SP
Book Reference: T1, C9
Date Conducted: 3/8/18
Page No: 42

Public-key crypto systems:-



The Public key scheme can be used for either secrecy or authentication or both.

Security:-

There is some source A produces a message in plain text $X = [x_1, x_2 \cdots x_m]$

- The message is intended for destination B.

B generates pair of keys: a public key PU_b & Private key PR_b where PU_b is publicly available.

- A will encrypt the message with PU_b key are generate cipher text Z and transmit to B. $Z = E(PU_b, X)$
- B will recive the cipher text and use the private key PR_b to decrypt the message. $X = D(PR_b, Z)$

Authentication:-

- A generates pair of key's: Public key PU_a & Private key PR_a where PU_a is publicly available, PR_a is only with A.
- A will encrypt message with Private key PR_a and output cipher text transmit to B $Y = E(PR_a, X)$
- B will recive message and decrypt with A's public key PU_a

Applications of Public-key cryptosystem:-

Application of Public-key cryptosystem can be classify in to three categories:-

- Encryption/decryption:- The sender encrypts a message with recipient's public key.
- Digital signature:- The sender "signs" a message with its Private key. signing is achieved by a cryptographic algorithm. applied to the message.
- Key Exchange:- Two side cooperate to exchange a session key.

Requirement for public key cryptography:-

1. It is computationally easy for a party B to generate a pair (PU_b, PR_b)
2. It is computationally easy for a sender A, knowing the public-key and message to be encrypted, M, to generate ciphertext.

$$C = E(PU_b, M)$$

3. It is computationally easy for the reciver B to decrypt the resulting ciphertext using the private key to recover the original message.

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

4. It is computationally infeasible for an adversary, knowing the Publickey, PU_b, to determine the private key, PR_b.
5. It is computationally infeasible for an adversary, knowing the Publickey, PU_b, ciphertext C, to recover the original message M.
6. The two keys can be applied in either order:

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E[PU_b, M)]$$

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Diffie-Hellman key exchange

Unit No: III
Lecture No: L28
Link to Session Planner (SP):S.No. 33 of SP
Book Reference: T1/C9
Date Conducted: 1/9/18
Page No: 43

The first published public-key algorithm by Diffie & Hellman
The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of message.

This algorithm itself is limited to the exchange of secret values.
The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.

The Algorithm.

Global Public Element

q — q - is prime number

α — $\alpha < q$ and α a primitive root of q.

User-A Key Generation

Select private X_A — $X_A < q$

Calculate public Y_A — $Y_A = \alpha^{X_A} \bmod q$.

User-B Key Generation

Select Private X_B — $X_B < q$.

Calculate public Y_B — $Y_B = \alpha^{X_B} \bmod q$.

Calculation of Secret key by user A & B

User-A

$K = (Y_B)^{X_A} \bmod q$.

User-B

$K = (Y_A)^{X_B} \bmod q$.

Summaizes Diffie-Hellman Key Exchange algorithm:

- There are two Publicly know number's: Prime number q & an integer that is a primitive root of q.

User A select a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A} \bmod q$.

User B select a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \bmod q$.

Each side keep's the X value private and Y value publicly to the other side.

User A computes the key as $K = (Y_B)^{X_A} \bmod q$.
User B computes the key as $K = (Y_A)^{X_B} \bmod q$. } equal

$$K = (Y_B)^{X_A} \bmod q$$
$$= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$$
$$= (\alpha^{X_B})^{X_A} \bmod q \quad \therefore \text{ by the rules of modular arithmetic}$$
$$= (\alpha^{X_A})^{X_B} \bmod q$$
$$= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$$
$$= (Y_A)^{X_B} \bmod q.$$

The ~~se~~ result is that two sides have exchanged a secret value.

Ex $q = 23$ then $\alpha = 5$

A
$X_a = 6$ then $Y_a = 5^6 \bmod 23$
$Y_a = 8$

B select secret key
$X_b = 15$ the $Y_b = 5^{15} \bmod 23$
$Y_b = 19$

$K = 19^6 \bmod 23 = 2$

$K = (8)^{15} \bmod 23 = 2$

$K = 2$

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Diffie-Hellman.

Unit No:	III
Lecture No:	L28
Link to Session Planner (SP): S.No. of SP	33
Book Reference:	T1/C9
Date Conducted:	11/9/18
Page No:	44

Key Exchange protocols.

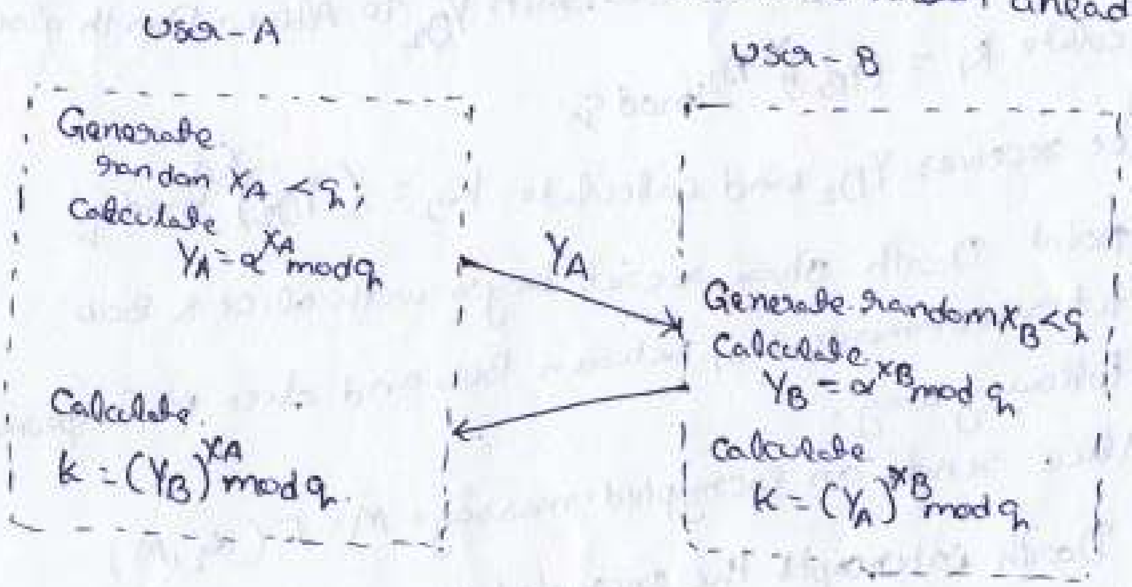
It is a simple protocol that makes use of Diffie-Hellman calculation.
user A wish to setup a connection with user B and use a secret key to encrypt messages on that connection.

→ User A can generate a one-time private key X_A, calculate Y_A and send that to the user B.

→ User B responds by generating a private value X_B calculate Y_B and send that to the user A.

→ Both users can calculate a key.

→ the values q and α would need to be known ahead of time



Man in the Middle Attack

The protocol is insecure against a man in the middle Attack.
If Alice and Bob wish to Exchang key. Darth is the adversary. The attack processed as follows

1. Darth prepares for attack by generating two random private keys X_{D_1} and X_{D_2} and then computing the corresponding Y_{D_1} and Y_{D_2}
2. Alice transmit Y_A to Bob
3. Darth intercepts Y_A and transmits Y_{D_1} to Bob. Darth also calculate $K_2 = (Y_A)^{X_{D2}} \bmod q$
4. Bob receives $Y_{D'}$ and calculate $K_1 = (Y_{D'})^{X_B} \bmod q$
5. Bob transmits Y_B to Alice.
6. Darth intercepts X_A and transmits Y_{D_2} to Alice. Darth also calculate $K_1 = (Y_B)^{X_{D_2}} \bmod q$
7. Alice receives Y_{D_2} and calculate $K_2 = (Y_{D_2})^{X_A} \bmod q$

At this point Darth share secret key's with alice & Bob. All the future communication between Bob and alice is Compromised in the following way.

1. Alice sends an encrypted message $M = E(K_2, M)$
2. Darth intercepts the encrypted message and decrypts it
3. Darth sends Bob $E(K_1, M)$ or $E(K_1, M')$, where M' any msg.

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: RSA Algorithm

Unit No:	III
Lecture No:	L29
Link to Session Planner (SP):S.No:... of SP	34
Book Reference:	T, K9
Date Conducted:	5/9/14
Page No:	45

Ron Rivest, Adi Shamir, and Len Adleman published RSA (Rivest shamir, Adleman) at MIT in 1978.

The RSA scheme is a block cipher in which the plain text and cipher text are integers between 0 and n-1 for some n.

Description of the Algorithm.

Key Generation

Select p, q — P and q both prime, $P \neq q$

Caculate $n = P \times q$

caculate $\phi(n) = (P-1)(q-1)$

Select integer e — $\gcd(\phi(n), e) = 1$; $1 < e < \phi(n)$

Calculate d — $de \bmod \phi(n) = 1$

Public key — $KU = \{e, n\}$

Private key — $KR = \{d, n\}$

Encryption

Plain text: $M < n$

Ciphertext $C = M^e (\bmod n)$

Decryption

Ciphertext C

Plain text $P = C^d (\bmod n)$

→ Both sender and receiver must known the value of n and e. and only receiver knows the value of d.

So, Public key encryption with Public key $KU = \{e, n\}$ Private key $KR = \{d, n\}$

For this algorithm to be satisfactory for public key encryption the following requirements must be met.

1. It is possible to find values of e, d, n such that $M^{ed} \bmod n = M$ for all $M<n$.

2. It is relatively easy to calculate M^e mod and C^d for all values of $M<n$.

3. It is infeasible to determine d given e and n.

The Algorithm begins by selecting two prime number's P & q, and calculating their product n, which is the modulus for encryption and decryption.

find quantity $\phi(n)$, referred to as the Euler totient of n

Then select an integer e that is relatively prime to $\phi(n)$

Finaly calculated d as multiplicative inverse of e, modulo $\phi(n)$

Encryption :- $C = M^e \bmod n$; decryption:- $M = C^d \bmod n$.

Ex

1. select two prime No:
   $P = 17$ and $q = 11$

2. calculate $n = p \times q$
   $17 \times 11 = 187$

3. Calculate $\phi(n)$.
   $(P-1)(q-1) = 16 \times 10 = 160$

4. Select a integer $\gcd(e, \phi(n)) = 1$.
   $e = 7$

5. Determine d such that
   $de \bmod 160 = 1$ and $d < 160$

• $23 \times 7 = 161 = (1 \times 160) + 1$

Public key $P_U = \{7, 187\}$
Private key $P_R = \{23, 187$

If plain text $M = 88$

* Encryption

$C = M^e \bmod n$
$= (88)^7 \bmod 187 = 11$
$C = 11$

• Decryption

$M = C^d \bmod n$
$M = 11^{23} \bmod 187$
$m = 88$

The Security of RSA

Four possible approaches to attacking the RSA algorithm.

→* Brute force :- This involves trying all possible private keys.

→* Mathematical attack :- There are several approaches, all equivalent in effort of factoring the product of two primes.

→* Timing attack :- These depend on the running time of the decryption algorithm.

→* Chosen ciphertext attack :- This type of attack exploits properties of the RSA algorithm.

* defense against the brute-force attack is using large key space

* Factoring problem :-

identify three approaches to attacking RSA mathematically.

* Factor n into its two prime factors. This enables calculation of $\phi(n) = (P-1)\times(q-1)$ turn, enables find $d = e^{-1} (\mathrm{mod}\ \phi(n))$

* Determine $\phi(n)$ directly, without first determining P and q. $d = e^{-1}(\mathrm{mod}\ \phi(n))$.

* Determine d directly, without first determining $\phi(n)$.

To avoid value of n that may be factored more easily the algorithm inventors suggest the following constraints P & q.

1. P and q should differ in length by only a few digits.
2. Both (P-1) and (q-1) should contain a large prime factor.
3. gcd (P-1, q-1) should be small.

→ Timing Attacks :-

Paul kocher, a cryptographic consultant, demonstrated that a snooper can determine ~~a storming one~~. a private key by keeping track of how long a computer takes to decipher messages. Although the timing attack is a serious threat. there are simple countermeasures that can be used, including the following

* Constant Exponentiation time :- Ensure that all Exponentiations ~~key~~ take the same amount of time before returning a result.
* Randomdelay :- Better performance could be achieved by adding a random delay to the Exponentiation algorithm to confuse the timing attack.
* Blinding :- Multiply the ciphertext by a random number before, performing exponentiation. This process prevents the attacker from knowing what ciphertext bits are being processed.

→ Chosen Ciphertext Attack

The basic RSA algorithm is vulnerable to a chosen cipher Attack (CCA). CCA is defined as an attack in which adversary chooses a number of ciphertext and is then given the corresponding plaintext, decrypted with the target's private key.

the adversary could select a plaintext, encrypt it with the ~~too~~ target's public key and the ~~be~~ able to get the plaintext back by having it's decrypted with the private key

$$E(PU, M_1) \times E(PU, M_2) = E(PU, M_1 \times M_2)$$

1. Compute $X = (C \times 2^e) \bmod n$
2. Submit X as a chosen ciphertext and receive back $Y = X^d \bmod n$

$$X = (C \bmod n) \times (2^e \bmod n)$$
$$= (M^e \bmod n) \times (2^e \bmod n) = (2M)^e \bmod n$$

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Key Management:

Unit No: III
Lecture No: L30
Link to Session Planner (SP): S.No. 35 of SP
Book Reference: T1/C10
Date Conducted: 6/9/11
Page No: 43

→ Public-key encryption helps address key distribution problem.

→ It has have two aspects of this:

→ Distribution of public key's for public-key encryption.

→ Use of public-key encryption to distribute secret keys.

Distribution of Public key's

It can be considered as using one of:

* Public announcement
* Publicly available directory
* Public-key authority
* Public-key certificates

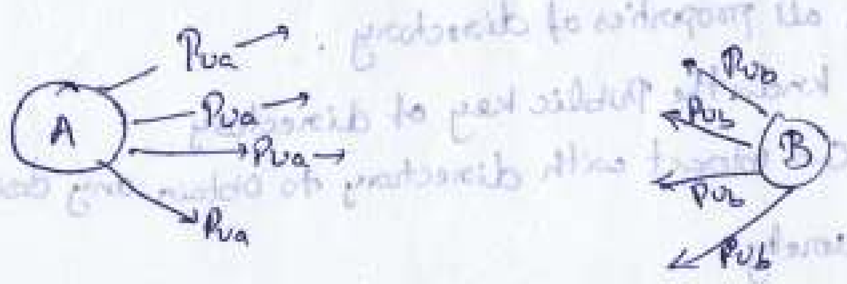
→ Public announcement:

Users distribute public keys to recipients or broadcast to community at large.

* append PGP key's to email messages or post

→ Its major weakness is forgery, anyone can create a key claiming to be someone else and broadcast it.

* Until forgery is discovered can masquerade as claimed user.



Publicly Available Directory

A greater degree of security be registering key's with a public directory.

→ Directory must be trusted with properties :-

1. Authority maintains a directory with a {name, public key} entry for each participant.
2. Each participant registers a public key with the directory authority.
3. Participent can replace key at any time
4. directory is periodically published.
5. directory can be accessed electonically

→ This secheme is clearly more secure than public announcements but still has vulnerabilities.

If an adversary succeeds in obtaining or computing the Private key of the directory authority. he can change any public key.

Public-key Authority :-

It Improve security by tightening control over distribution of key's from directory.

It has all properties of directory.

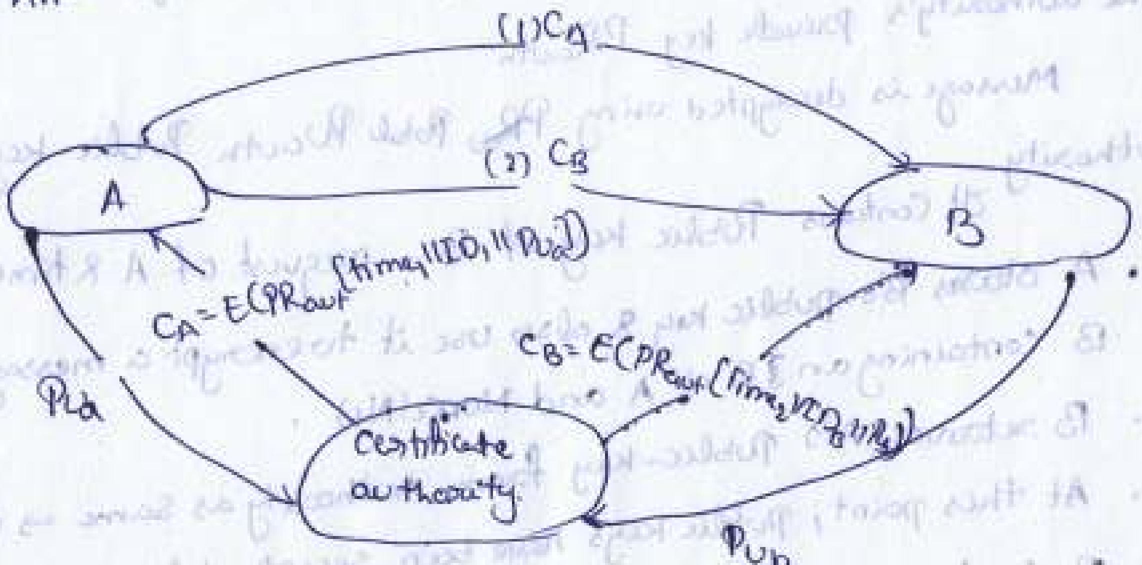
→ User must know the public key of directory then user can interact with directory to obtain any desired public key securely.

Public-Key certificates:-

→ Certificates allow key exchange without real-time access to Public-key authority.

→ A certificate binds identity to public key, with all Contents such as period of validity, rights etc. Signed by a trusted Public-key or Certificate Authority (CA) and it can be verified by anyone who knows the public-key authorities public-key.

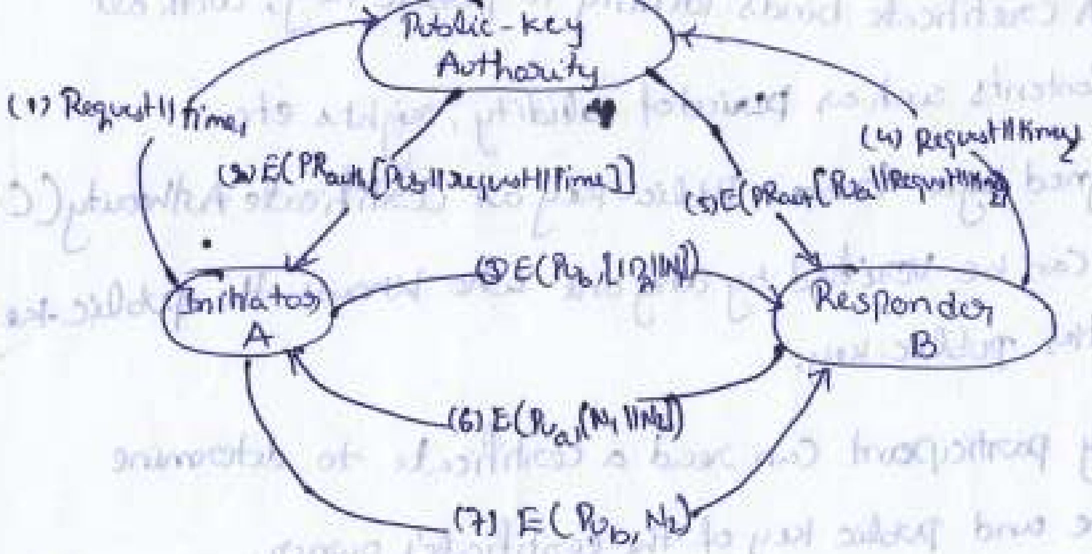
1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
2. Any Participant can verify that certificate originated from the Certificate authority and is not counterfeit.
3. Only the Certificate authority can create and update certificates.
4. An



Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Key management

Unit No:	III
Lecture No:	L30
Link to Session Planner (SP): S.No: ... of SP	35
Book Reference:	T1 C10
Date Conducted:	6/9/18
Page No:	48

Public-key Authority.



Steps

1. A sends a timestamp message to the public-key authority as a request for the current public key B.
2. The authority responds with a message that is encrypted using the authority's private key PR_{auth}

   Message is decrypted using ~~PR~~ ~~PUb~~ PU_{auth} Public-key of Authority.

   It contains Public key of B, request of A & timestamp.
3. A stores B's public key & also use it to encrypt a message to B containing an ID of A and Nonce(N_1).
4. B retrieves A's public-key from Authority as same as A - B.
5. At this point, public keys have been securely delivered to A and B.
6. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as new nonce N_2
7. A returns N_2, encrypted using PU_b to ~~ens~~ assure B that its correspondent is A

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Key Management :3

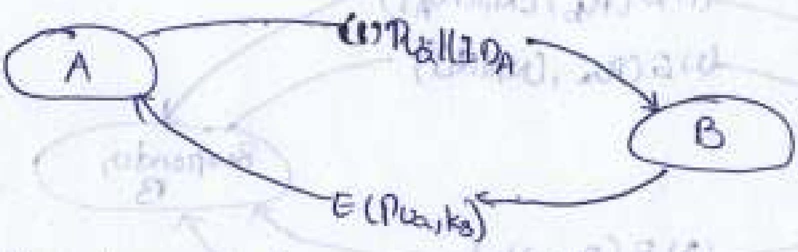
Unit No: III
Lecture No: L80
Link to Session Planner (SP): S.No. 35 of SP
Book Reference: T1 | 90
Date Conducted: 6/9/11
Page No: 49

Distribution of Secret keys using Public-key Cryptography:-

Simple Secret Key Distribution:-

An If A wishes to Communicate with B,

1. A generates a public/private key pair (PU_a, PR_a) and transmits a message to B consisting of PU_a and identifier of A, ID_A.
2. B generates a secret key, K_s and transmits it to A, encrypted with A's public key
3. A computes $D(PR_a, E(PU_a, K_s))$ to recover the secret key.
4. A discards PU_a and PR_a and B discards PU_a.



This is not secure for man-in middle attack.

(1) A generates a Public/Private key pair {PU_a, PR_a} and transmits a message intended for B consisting of PU_a and an identifier of A, ID_A.
(2) E intercepts the message, creates its own Public/private key pair {PU_e, PR_e} and transmits $PU_e \| ID_A$ to B.
(3) B generates a secret key, K_s and transmits $E(PU_e, K_s)$.
4) E intercepts the message, and learns K_s by computing $D(PR_e, E(PU_e, K_s))$
5) E transmits $E(PU_a, K_s)$ to A.

Secret key Distribution with Confidentiality and authentication

Then the following step's occur:

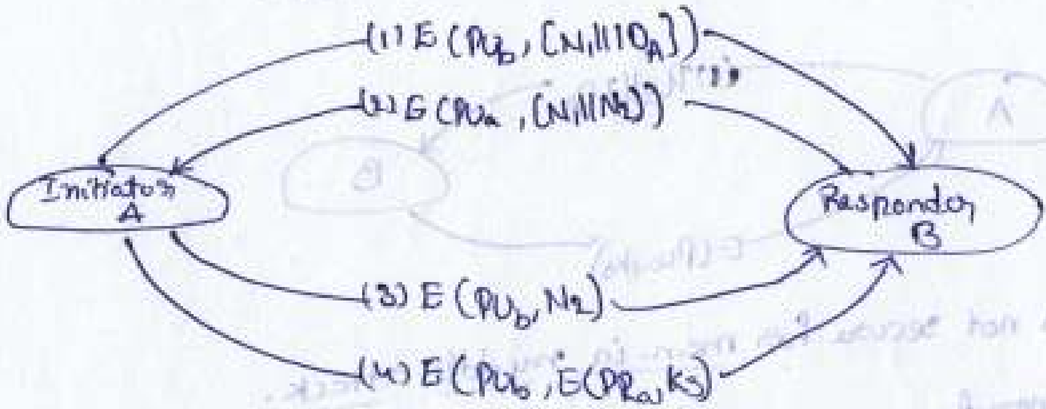
1. A uses B's public key to encrypt a message to B containing an identifier of A(ID_A) and nonce (N_1) which is used to identify this transaction uniquely.

2. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B(N_2)

3. A returns N_2 encrypted using B's public key, to assure B that its correspondent is A.

4. A selects a secret key K_s and sends $M = E(PU_b, E(PR_a, K_s))$ to B.

5. B computes $D(PU_a, D(PR_b, M))$ to recover the secret key.



Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Elliptic Curve Arithmetic

Unit No:	III
Lecture No:	L31
Link to Session Planner (SP):S.No....of SP	36
Book Reference:	T1, 90
Date Conducted:	8/9/18
Page No:	20

The principal attraction of ECC, compared to RSA, is that it appears to offer equal security for a smaller key size, thereby reducing processing overhead.

The confidence level in ECC is not yet high as that in RSA.

Abelian groups:

An abelian group G, sometimes denoted by $\{G, \cdot\}$, is a set of elements with binary operator, denoted by $\cdot$ that associates to each ordered pair (a, b) of elements in G an element $(a \cdot b)$ in G, such that the following axioms are obeyed.
[generic refers to addition, multiplication or other]

(A1) Closure: If a and b belong to G, then $a \cdot b$ is also in G.

(A2) Associative: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all a, b, c in G.

(A3) Identity element: There is an element e in G such that $a \cdot e = e \cdot a = a$ for all a in G

(A4) Inverse element: For each a in G there is an element a' in G such that $a \cdot a' = a' \cdot a = e$

(A5) Commutative: $a \cdot b = b \cdot a$ for all a, b in G

A number of public key ciphers are based on the use of an abelian group.

Ex: Diffie-Hellman: $a^k \bmod q = \underbrace{(a \times a \times a \cdots \times a)}_{k \text{ times}} \bmod q$

For elliptic curve cryptography, an operation over elliptic curves called addition, is used.

$$a \times k = \underbrace{(a + a + a + \cdots a)}_{k \text{ times}}$$

Elliptic curves over real Numbers:-

Elliptic curves are not ellipses. They are so named because they are described by cubic equation, similar to those used to calculating the circumference of an ellipse. In general, Cubic equations for elliptic curves take the form.

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

where a, b, c, d and e are real numbers and x and y take on values in real numbers.

Equation:- elliptic curve → $y^2 = x^3 + ax + b$.

Such equation said to be cubic, or degree 3, because the highest exponent they contain is a 3.

Also included in the definition of a elliptic curve is a single element denoted O and called the point at infinity or the zero point.

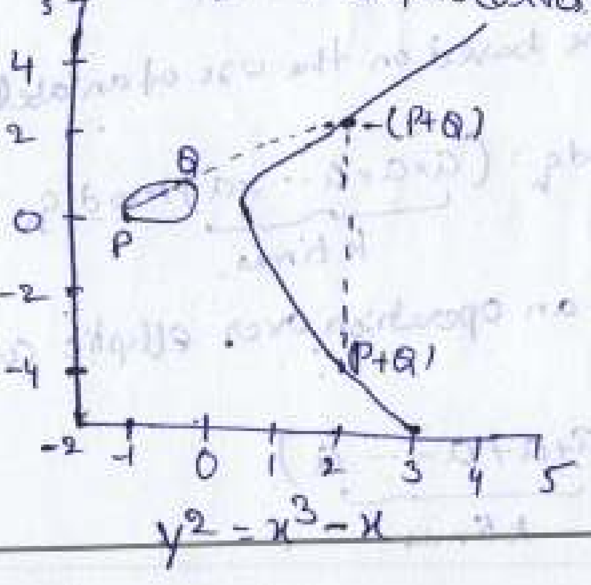
To plot such a curve, compute:

$$y = \sqrt{x^3 + ax + b}$$

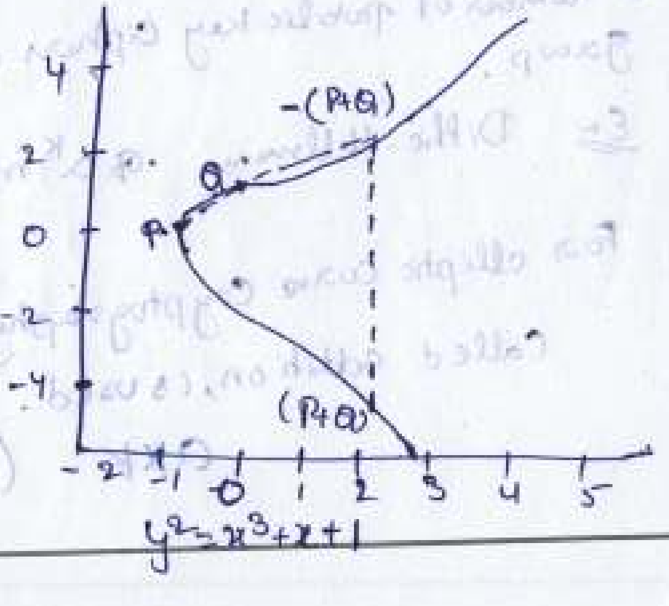
→ for a given value of a and b, the plot consists of positive and negative values of y for each value of x.

Ex Thus each curve is symmetric about y=0.

two Example of elliptic curves



$y^2 = x^3 - x$



$y^2 = x^3 + x + 1$

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: ECC - A

Unit No: III
Lecture No: L51
Link to Session Planner (SP):S.No.86 of SP
Book Reference: T1/C10
Date Conducted:
Page No: 5

Now, Consider the set of point $E(a, b)$ Consisting of all of the points (x, y) that statisfy together with the element O. Using a different value of the pair (a, b) result in a different set $E(a, b)$

Elliptic Curves over Z_p

Elliptic curves cryptography makes use of elliptic curves in which the variables and coefficients are all restricted to elements of a finite field.

Two families of elliptic curves are used in cryptographic application: Prime curves over Z_p
Binary curves over $GF(2^m)$.

→ For Prime curves over Z_p, uses a cubic equation in which the variables and coefficients all take on values in the set of integers from 0 to $p-1$ and in which calculations are performed over modulo p.
* best for software Application

→ For Binary curves defined over $GF(2^m)$, the variables and coefficients all take on values in $GF(2^m)$ and calucations are performed over $GF(2^n)$
* hardware Application

for elliptic curves over Z_p, Equation.

$$y^2 \bmod p = (x^3 + ax + b) \bmod p$$

Elliptic Curves over GF(2^m)

A finite field GF(2^m) consists of 2^m elements. together, with addition and multiplication operations that can be defined over Polynomials.

For Elliptic Curves over GF(2^m) is use cubic equation.

$$y^2 + xy = x^3 + ax^2 + b$$

where ithe variable x and y and the coefficient a and b are elements of GF(2^m) of and that calculations are performed in GF(2^m).

1. the addition operation in ECC is the counter part of modular multiplication in RSA
→ Multiple addition is the counterpart of modular exponentiation.
→ To form a cryptographic system using elliptic curves, Need to find a "hard problem" corresponding to factoring of two primes or taking the discrete logarithm.
→ Consider the equation $Q = kP$, where Q, P belong prime curve $Q, P \in E_p(a,b)$ $k < p$
- It is "easy" to compute Q given k, P.
- but "hard" to find k given Q, P

– This is called the discrete logarithm problem for elliptic curves.

Ex: Consider a group $E_{23}(9,17)$. This group defined by equation $y^2 \bmod 23 = (x^3 + 9x + 17) \bmod 23$

ECC Diffie-Hellman key Exchange

A key Exchange between user A and B can be accomplished.

1. A select an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A \times G$; the public key point is $E_q(a,b)$
2. B similarly selects a private key n_B and compute P_B.
3. A generate the secret key $K = n_A \times P_B$. B generates secret key $K = n_B \times P_A$

Two Calculation on step 3 produce the same result because

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$$

Global Public Elements

(1) $E_q(a,b)$ elliptic curve with parameters a, b and q, where q is a prime or an integer of the form 2^m.

(2) G point on elliptic curve whose order is large value n.

User A key Generation

(1) select Private n_A $n_A < n$

Calculate public P_A $P_A = n_A \times G$.

User B Key Generation

Select private $n_B < n$.

Calculate public P_B $P_B = n_B \times G$

Calculate Secret key by user A & B.

User A

$K = n_A \times P_B$

User B

$K = n_B \times P_A$

To Break this schem, an attacker need to be able to Compute K given G and KG, which is assumed hard.

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Elliptic Curve Encryption

Unit No: III
Lecture No: 32
Link to Session Planner (SP): S.No 37 of SP
Book Reference: T1
Date Conducted: 8/9/10
Page No: 53

Elliptic Curve Encryption/Decryption

→ First task in this system is to encode the plaintext message 'm' to be sent as an x-y point P_m.

It is the point P_m that will be encrypted as a ciphertext and subsequently decrypted.

Then key Exchange system has to share public key.

To encrypt and send a message P_m to B,

A chooses a random positive integer k and produces ciphertext C_m consisting of the pair of points

$$C_m = \{kG, P_m + kP_B\}$$

Note: A has used B's public key P_B.

To decrypt the ciphertext, B multiplies the first point in the pair by B's secret key and subtracts the result from second point.

$$P_m + kP_B \; n_B(kG) = P_m + k(n_B G)\, n_B(kG) = P_m$$

Security of Elliptic Curve Cryptography

The security of ECC depends on how difficult is to determine k, given kP and P. which is elliptic curve logarithm problem.

The fastest known technique for taking the elliptic curve logarithm is known as the "Pollard rho method"

Comparable key Size in terms of Computational Effort for Cryptanalysis.

Symmetric scheme. (key size)	ECC - Based Scheme. (key size)	RSA/DSA (key size)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Authentication Function:-

Unit No:	III
Lecture No:	L34
Link to Session Planner (SP):S.No... of SP	39
Book Reference:	T1 / C-11
Date Conducted:	19/9/18
Page No:	54

Types of functions that may be used to produce an authenticator:- these may be grouped into three classes.

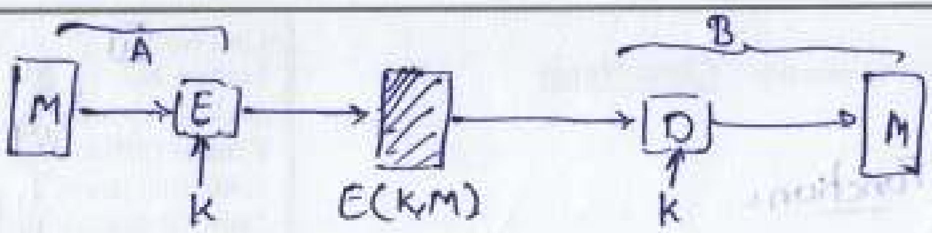
* Message encryption:- The ciphertext of the entire message serves as its authenticator.
* Message authentication code (MAC): A function of the message and secret key that produces a fixed-length value that serves as the authenticator.
* Hash function:- A function that maps a message of any length into fixed-length hash value, which serves as the authenticator.

* Message authentication is concerned with:-
    * Protecting the integrity of message
    * Validating identity ~~of~~ of originator
    * non-repudiation of origin (dispute-resolution)

Message Encryption :-

Message Encryption by itself also provides a measure of authentication

→ If Symmetric encryption is used then:-
Only sender and receiver are sharing the secret key.
→ receiver know sender must have created of message.
→ It is in Encrypted format Content cannot been altered.

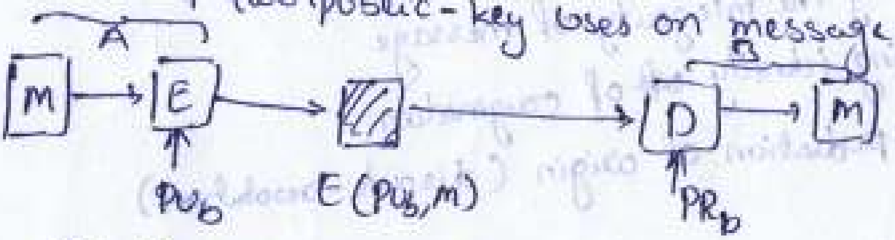


Symmetric encryption:- Confidentiality & authentication.

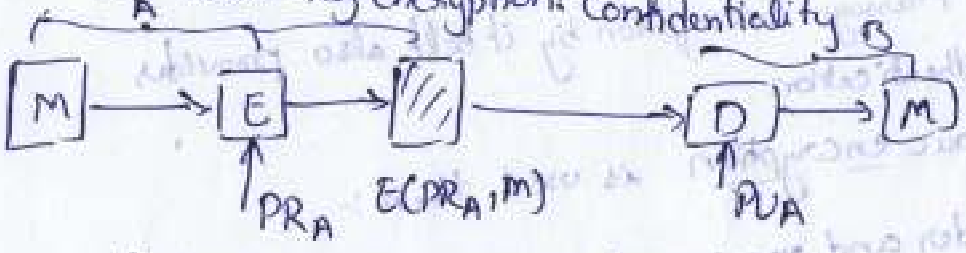
If Public-key encryption is used :-

↳ Encryption provides no confidence of sender.
because any one can Encrypt the message using public-key.
Other way to provide Autentication as.
↳ sender sign's message using their private-key of sender then encrypts with recipients public key.
It will have both ~~security~~ security and authentication.

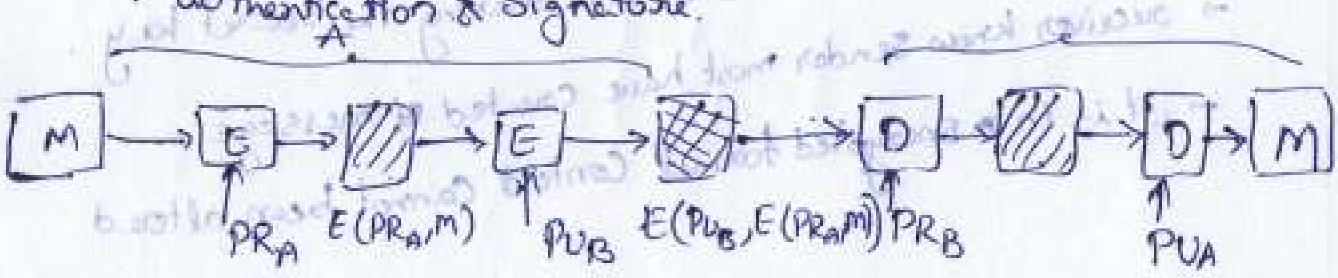
* Need to recognize corrupted message.
* It cost of two public-key uses on message



(b) Public-key encryption: Confidentiality



: authentication & signature.



Subject: Cryptography and network security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Authentication function

Unit No: III
Lecture No: L3.1
Link to Session Planner (SP): S.No. 39 of SP
Book Reference: T1/C11
Date Conducted: 19/9/18
Page No: 55

(2). Message Authentication Code (MAC)

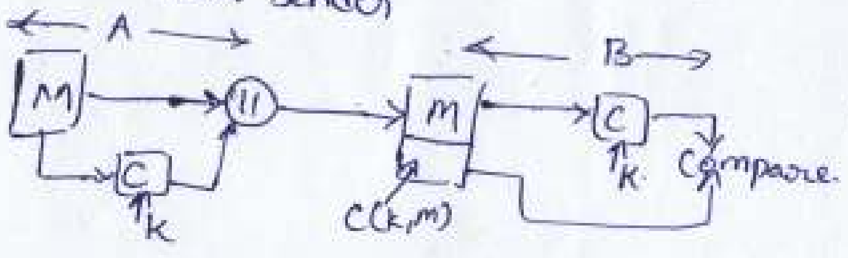
An alternative authentication technique involves the use of a secret key to generate a small fixed-size of block of Data known a Cryptographic Checksum (or) MAC that is appended to message.

MAC block is depending on both message and key. This technique assume that two communicating parties, A and B sharing a common key K. A MAC function is similar to encryption Except that the MAC algorithm need not be reversible. MAC is appended to message as a signature.

→ A Create's a MAC and appended at the end of Message and transmits to B.

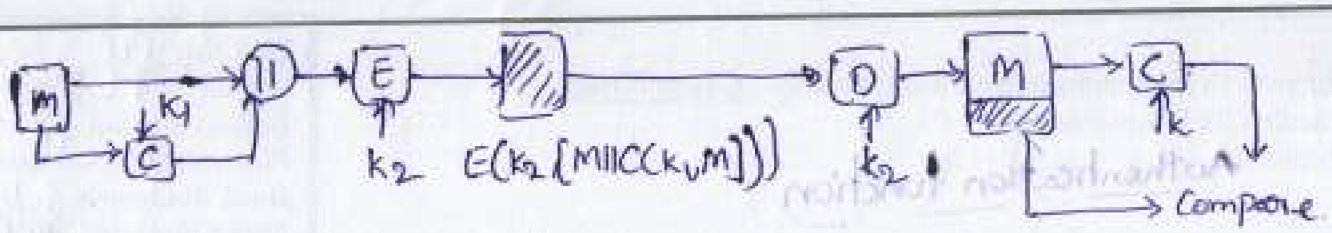
→ B recive's the message and MAC. Now B will create the MAC on message and Compare with recived MAC if it matches the MAC

It Provides assurance that message is unaltered and Comes from sender

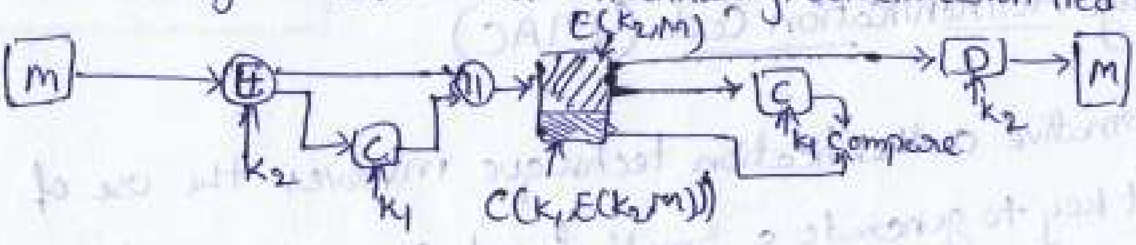


∴ C -> MAC function

Message Autentication



(b) Message authentication & confidentiality: authentication tied to plain text.



(c) Message authentication & confidentiality: authentication tied to ciphertext

③ Hash Function :-

A Variation on the message authentication code is the one-way hash function.

As with the Message authentication, a hash function accepts a variable size message M as input and produces a fixed-size output referred to as a hash code H(M).

$$h = H(M)$$

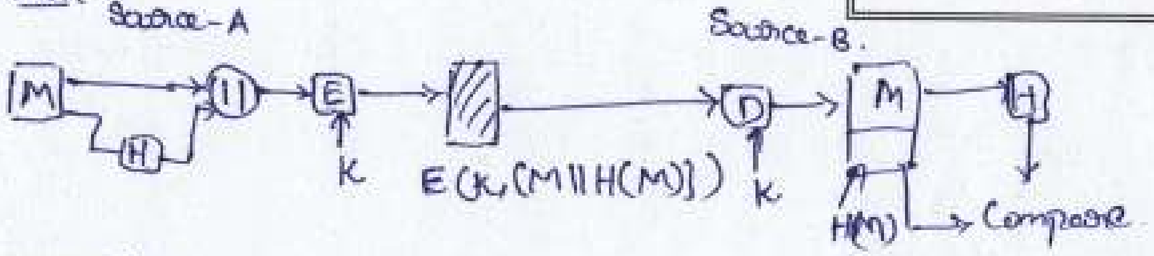
→ A hash code does not use a key but is a function only of the input message.

The hash code is also referred to as a message digest or hash value. This hash value is used to detect changes to message most often to create a digital signature hash is used.

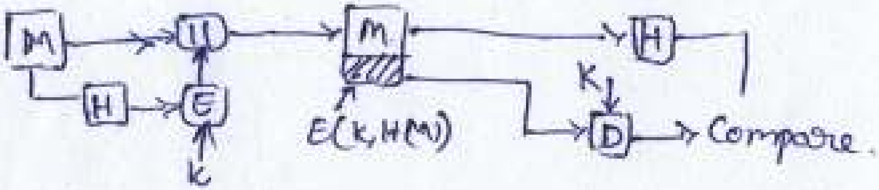
Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Authentication function.

Unit No:	III
Lecture No:	L34
Link to Session Planner (SP): S.No	39 of SP
Book Reference:	T1
Date Conducted:	19/9/18
Page No:	56

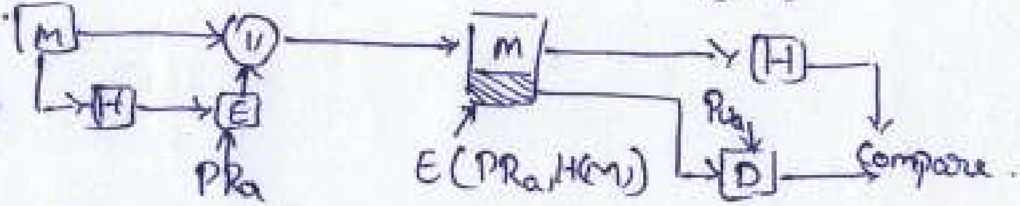
hash function



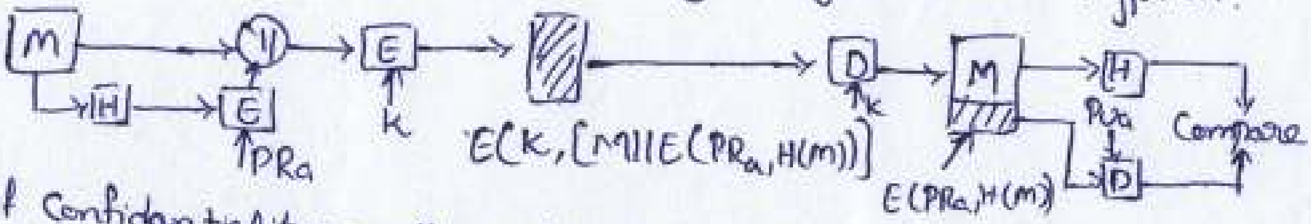
The message + hash code is encrypted using symetric encryption.



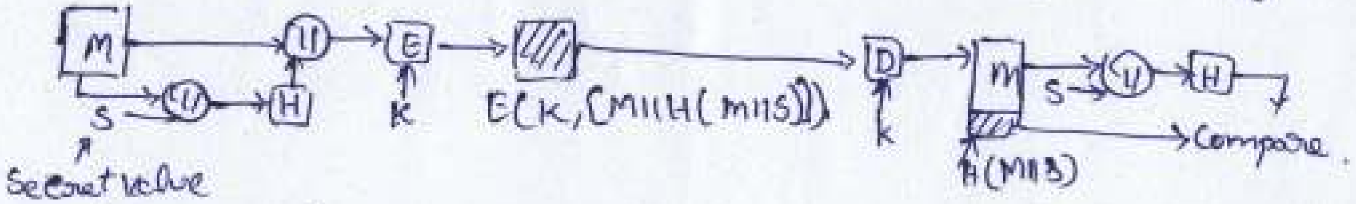
Only hash code is encrypted, using symmetric encryption.



only hash code is encrypted, using Asymmetric encryption.



If confidentiality as well as a digital signature is desired, then message plus the private key encryped hash code can be encrypted using symmetrik key.



Confidentiality can be added to the approach for Authentication code without Encrypting code.

17

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Security of Hash functions and MAC's

Unit No: III
Lecture No: L36
Link to Session Planner (SP): S.No. 41 of SP
Book Reference: T1 | C11
Date Conducted: 22/9/18
Page No: 57

* Attacks on A hash functions and MACs can grouped into two Categories:

· brute-force attacks and Cryptanalysis.

Brute-force attacks:-

The nature of brute-force attacks differ somewhat for hash functions and MAC's.

Hash function:-

The strength of a hash function against brute-force attacks depends solely on the length of the hash code produced by the algorithm.

hash function properties:-

* one-way:- For any given code h, it is computationally infeasible to find x such that $H(x) = h$

* weak collision resistance:- for any given block x, it is computationally infeasible to find $Y \neq X$ with $H(Y) = H(x)$.

· Strong collision resistance:- It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

· For a hash code of length n, level of effort required.

One way 2^n

weak collision resistance 2^n

strong collision resistance $2^{n/2}$

If strong collision resistance is required then the value $2^{n/2}$ determines the strength of the hash code against brute-force attacks.

* Oorschot & wiener presented a design for $10 million collision search machine for MD5, which has a 128-bit hash length. that could find collision in 24 days.

Message Authentication code :- A brute-force attack on a MAC is a more difficult undertaking because it requires known message - MAC pairs.

To attack a hash code.

• Given a fixed message x with n-bit hash code $h = H(x)$, a brute-force method of finding a collision is to pick a random bit string y and check if $H(Y) = H(x)$.

The attacker can do this repeatedly offline.

• Computation - resistance: Given one or more text - MAC pairs $[x_i, C(K, x_i)]$
- it is computationally infeasible to compute any text - MAC pair $[x, C(k, x)]$ for any new input $x \neq x_i$

If the attacker can determine the MAC key, then it is possible to generate a valid MAC value for any input x

An attacker can also work on the mac value without attempting to recover the key.

* The level of effort for brute-force attack on a MAC algorithm can expressed as $\min(2^k, 2^n)$.

The assessment of strength is similar to that for symmetric encryption algorithm. It would appear reasonable to require that the key length and MAC length satisfy a relationship such as $\min(k, n) \geq N$.

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Security of hash function & MAC's - 2

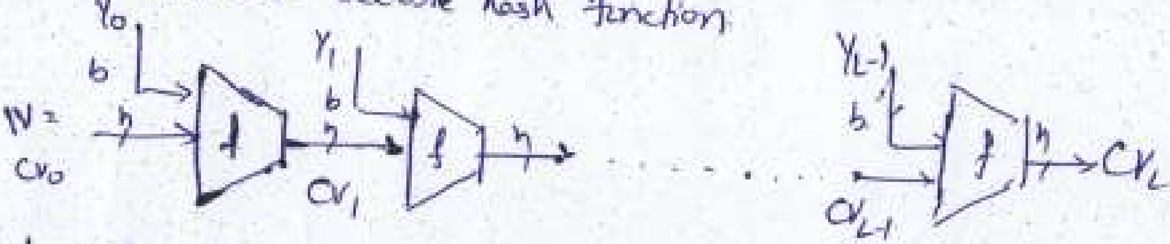
Unit No:	III
Lecture No:	L36M
Link to Session Planner (SP): S.No.... of SP	
Book Reference:	T1/C11
Date Conducted:	22/9/11
Page No:	58

Cryptanalysis :- Cryptanalytic attacks on hash-function and MAC algorithm seek to Exploit some property of the algorithm to perform some attack other than an exhaustive search.

- The way to measure the resistance of a hash or MAC algorithm to Cryptanalysis is to compare it's strength to the effort required for a brute-force attack

Hash Function :-

Structure of secure hash function



- The hash algorithm involves repeated use of a Compression function. f, that takes two inputs and produces an n-bit output.

CV_0 = IV = initial n-bit value

$CV_i = f(CV_{i-1}, Y_{i-1}) \quad 1 \le i \le L$

$H(M) = CV_L$

where the input to hash function is a message M consisting of the block $Y_0, Y_1, \ldots Y_{L-1}$

- Cryptanalysis of hash functions focuses on the internal structure of f and is based on attempts to find efficient techniques for producing collisions for single execution of f.

The attacker must take into account the fixed value of IV.

Massage Authentication code :- There is much more variety in the structure of MACs than in hash function., so its is difficult to generaleze about the cryptanlysis of MACs.

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Secure Hash Algorithm (SHA)

Unit No:	III
Lecture No:	37
Link to Session Planner (SP): S.No.	42 of SP
Book Reference:	1 / C12
Date Conducted:	22/9/11
Page No:	59

→ SHA originally designed by NIST & NSA in 1993
It was revised in 1995 as SHA-1.
It is based on design of MD4 with key diffrences.
SHA-1 produce 160-bit ~~sh~~ hash value.

Revised secure Hash standard:-
NIST issued revision FIPS 180-2 in 2002.
It adds 3 additional version of SHA.
SHA-256, SHA 384, SHA-512.
This are designed for Compatibility with increased security provided by the AES cipher.
Structure and detail is similar to SHA-1 but security levels are higher.

Comparison of SHA Parameters

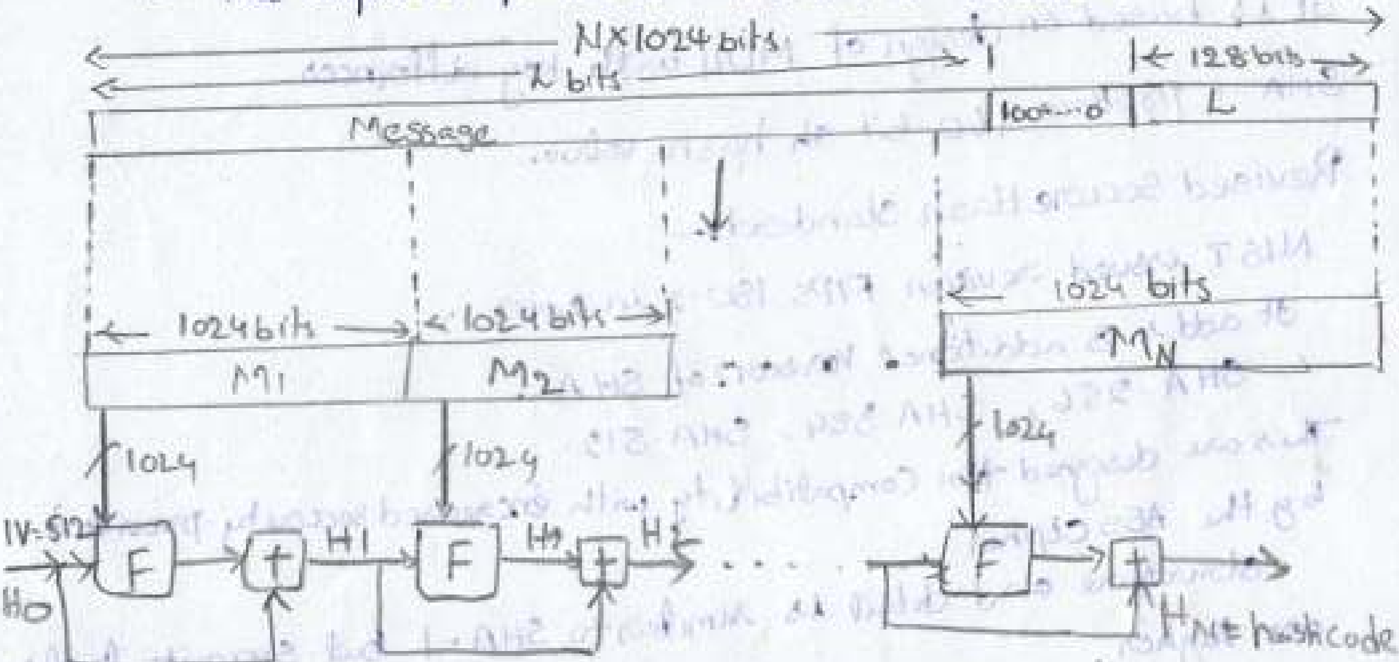
	SHA1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Block size	512	512	1024	1024
Message size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
word size	32	32	64	64
Number of steps	80	64	80	80
security	80	128	192	256

All size are measured in bits
2. Security refers to the fact that birthday attack on a message digest of size n. Produces a collision

SHA512 Logic :-

The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces, output a 512-bit message digest.

→ The input is produces in 1024 bit blocks.



+ ⇒ word by word addition mod 2^{64}

* step 1:- Append padding bits :- the message is padded so that its length is congruent to 896 modulo 1024.
Padding is always added, even if the message is already of desired length.
→ The number of padding bits is in the range of 1 to 1024
→ The padding consist of a single 1-bit, followed by necessary number of 0-bits.

* step-2 :- Append length :- A block of 128 bits is appended to the message. This block contains unsigned 128-bit integer which contains the length of the original message. (before padding)

outcome of this two steps yields message in to N×1024 bit size
It is represent as the sequence of 1024-bit blocks
$M_1, M_2, M_2 \cdots M_n$!

* step 3:- Initialize hash buffer

A 512-bit buffer is used to hold intermediate and final result of the hash function:-

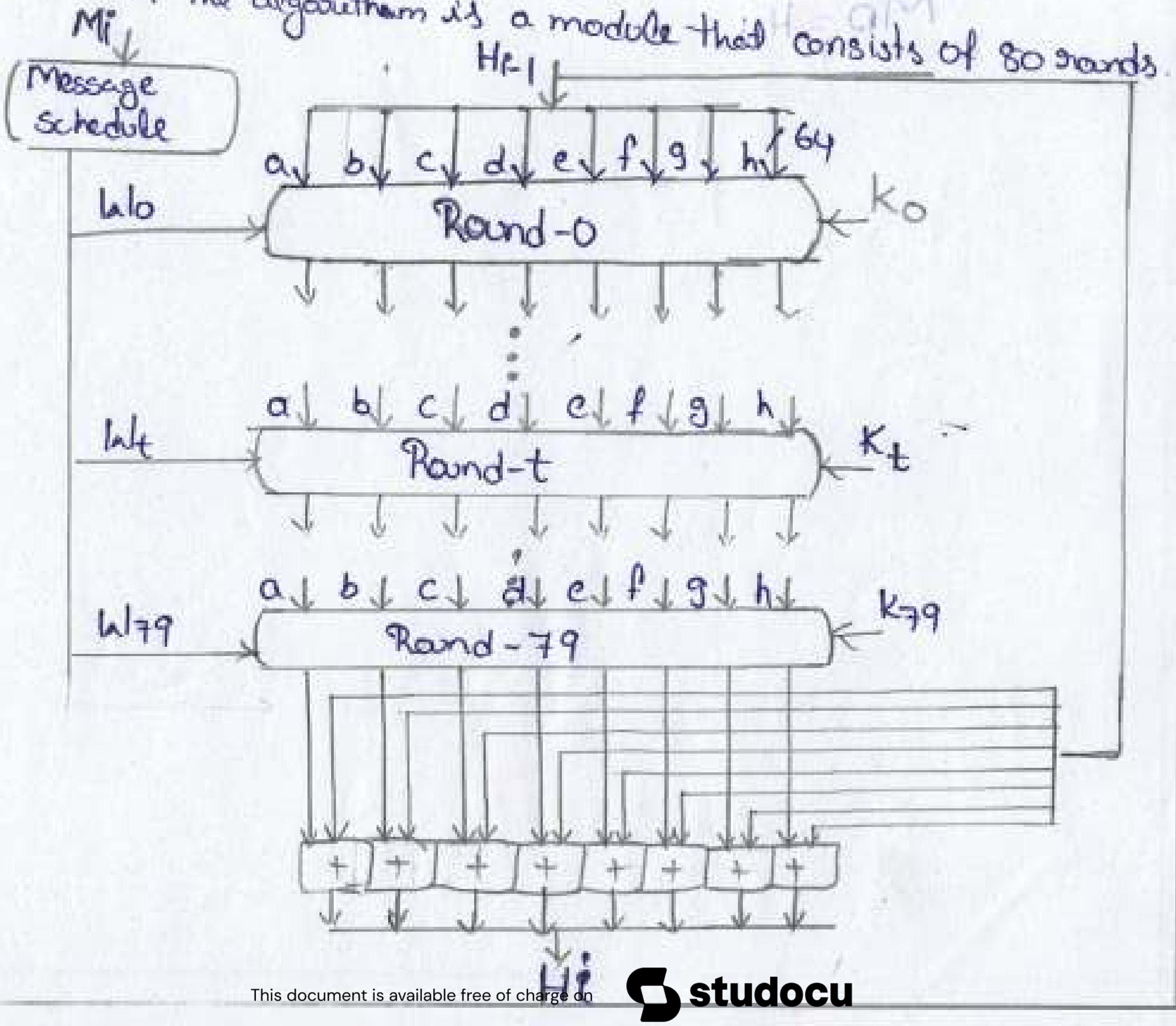
The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h) These registers are initialized to the following 64 bit integer (hexadecimal value):-

a = 6A09E667F3BCC908 b = BB67AE8584CAA73B c . . . .

These values are stored in big-endian format

* Step-4:- process message in 1024-bits (128-words) blocks:-

The heart of the algorithm is a module that consists of 80 rounds.



Each round takes a input of 512 bit buffer value abcdefgh. and update the contents of the buffer.

At the to the first round, the buffer has value of the intermediate hash value, H_{i-1}.

Each round t uses of a 64 bit value W_t derived from the current 1024-bit block being processed (M_i)

Step 5: Output:

After all N 1024-bit blocks have been processed, the output from the N^{th} stage is the 512-bit message digest.

$H_0 = IV$

$H_i = SUM_{64}(H_{i-1}, abcdefgh_i)$

$MD = H_N$

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Whirlpool

Unit No:	III
Lecture No:	L38
Link to Session Planner (SP): S.No.... of SP	43
Book Reference:	T_1/C12
Date Conducted:	26/9/18
Page No:	61

Whirlpool is only one of two hash function endorsed by NESSIE (New European Schemes for Signature, Integrity & Encryption) Project.

Whirlpool hash is based on the use of a modified AES block cipher as the compression function.
It is intended to provide security and performance that is comparable.

Whirlpool has following features.

1. The hash code length is 512 bits, equaling SHA.
2. The overall structure of the hash function is one that has been shown to be resistent to the usual attack on block cipher based hash code
3. The underlying block cipher is based on AES and is designed to provide for implementation in S/w and H/w

Whirlpool Hash Structure:-

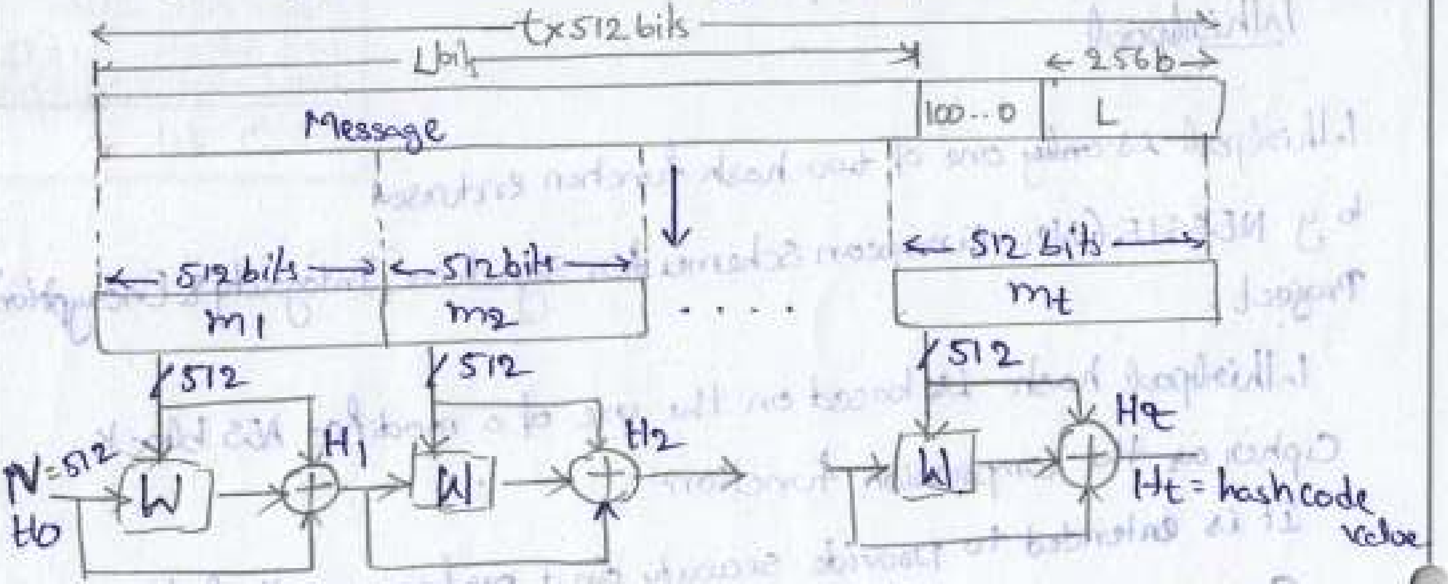
Whirlpool Logic:- Given message consisting of a sequence of blocks $m_1, m_2, m_3 \ldots m_t$ the whirlpool hash function is

H_0 = initial value

$H_i = E(H_{i-1}, m_i) \oplus H_{i-1} \oplus m_i$ = intermediate value

H_t = hash code value

Message Digest Generation Using whirlpool.



Step 1 :- Append padding bits :- The message is padded so that it length in bits is an odd multiple of 256.

→ Padding is always added, even if the message is alredy desired length. the number of padding bits is in a range of 1 to 512

Step 2 :- Append Length :- A block of 256 bits is appended to the message. This block is treated as unsinged integer of 256-bits and contains the length in bits of the original message the out come of two step's are a message of length t×512 bits. it is represent as $m_1, m_2, m_3 \ldots m_t$.

Step 3 :- Initialize hash matrix :- An 8×8 matrix of bytes is used to hold intermediate and final result of the hash function the matrix is initalized as consisting of all zero-bits

• ~~Process~~ Step 4 :- Process message in 512-bit (64-byte) block :- the heart of the algorithm is the blockcipher W.

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Whirlpool-2

Unit No: II
Lecture No: L38
Link to Session 43
Planner (SP): S.No:... of SP
Book Reference: T1/C12
Date Conducted: 26/9/18
Page No: 62

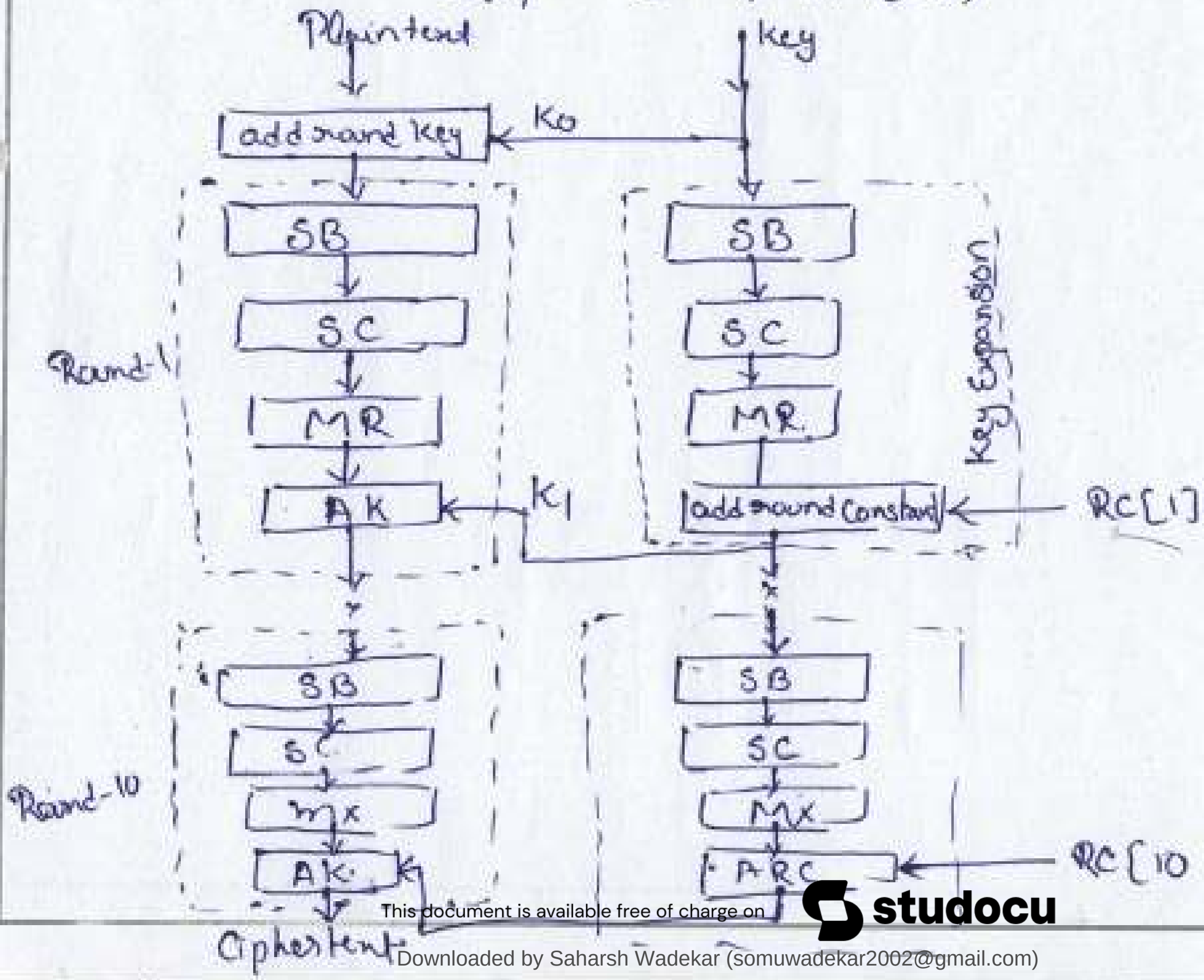
Block Cipher W.

The encryption algorithm takes a 512-bit block of Plain text and a 512 bit Key as input and produces a 512-bit block of cipher text as output.
The encryption involves the use of four different function. add key (AK), substitute bytes (SB), shift columns (SC) and Mix Row (MR).

W - Consist of single (AK) followed by 10 Rounds that involve all four functions.

$$RF(K_r) = AK[K_r] \circ MR \circ SC \circ SB.$$

$$W(K) = \left(\bigcirc_{r=1}^{10} RF(K_r) \right) \circ AK(K_0)$$



Whirlpool Performance

Whirlpool is a very new proposal, hence there is little Experience with use.

Many AES finding should apply

- Compare to SHA-512, whirlpool requires more hardware resources but performance much better in terms of throughput.

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: HMAC

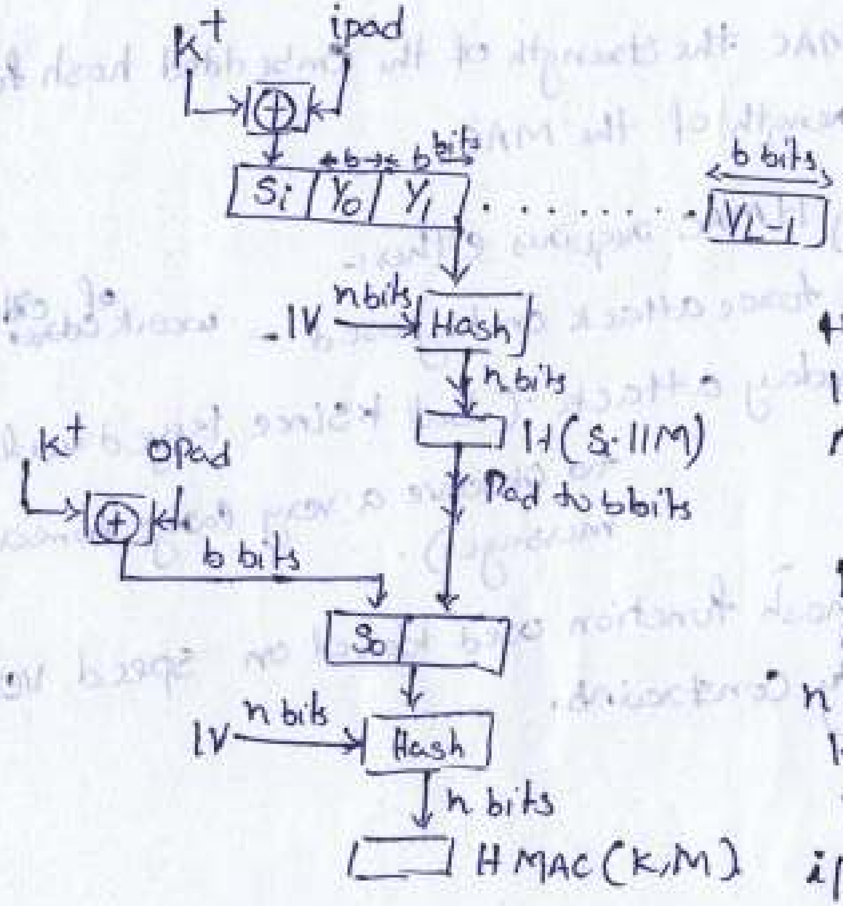
Unit No:	III
Lecture No:	L38
Link to Session Planner (SP):S.No... of SP	43
Book Reference:	T1/C12
Date Conducted:	26/9/18
Page No:	63

HMAC Design objectives

RFC 2104 lists the following design objective for HMAC.

- To use, without modifications, available hash functions.
- To allow for easy replaceability of the embedded hash function
- To preserve the original performance of the hash function without incurring a significant degradation
- To use and handle keys in a simple way
- To have a well understood cryptographic analysis of the strength of the authentication mechanism.

HMAC Algorithm:-



H → hash function (MD5, SHA)
IV - initial input to hash
M - Message input to hash
Y_i = i^{th} block of M
L = number of block
b = number of bits in a block
n = length of hash code
K = secret key
K^+ = K padded with zero's
ipad = 00110110 (36 in hex)
opad = 01011100 (5C in hex)

$$HMAC(K,M) = H[(K^+ \oplus opad) \| H[(K^+ \oplus ipad) \| (M)]]$$

In words:-

1. Append zero to the left end of K to create a b-bit String K^+

2. XOR K^+ with ipad to produce the b-bit block S_i

3. Append M to S_i

4. Apply H to the stream generated in step 3.

5. XOR K^+ with opad to produce the b-bit block S_o

6. Append the hash result from step 4 to S_o

7. Apply H to the stream generated in step-6 and output the result.

Security of HMAC:-

→ for any MAC the strength of the embedded hash function is the strength of the MAC.

→ attacking HMAC requires either:-

* brute force attack on key used. → work of order 2^n
* birthday attack (but since keyed would need to observe a very large number of message).

→ Choose hash function used based on speed verses security constrains.

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: CMAC

Unit No: III
Lecture No: L39
Link to Session
Planner (SP):S.No. 44 of SP
Book Reference: T1 / C12
Date Conducted: 27/9/18
Page No: 64

The Data Authentication Algorithm defined in FIPS PUB 113, also known as the CBC-MAC [cipher block-chaining message authentication code].

The operation of CMAC when the message is an integer multiple n of the cipher block length b.

For AES b = 128 and for triple DES, b = 64.

The message is divided into n blocks, $M_1, M_2 \ldots M_n$

The algorithm makes use of a k-bit encryption key K and an n-bit constant

- For AES key size K is 128, 192 or 256 bits
- for triple DES, the key size 122 or 168 bits.

CMAC is calculated as follows

$C_1 = E(K, M_1)$

$C_2 = E(K, [M_2 \oplus C_1])$

$C_3 = E(K, [M_3 \oplus C_2])$

$\vdots$

$C_n = E(K, [M_n \oplus C_{n-1} \oplus K_1])$

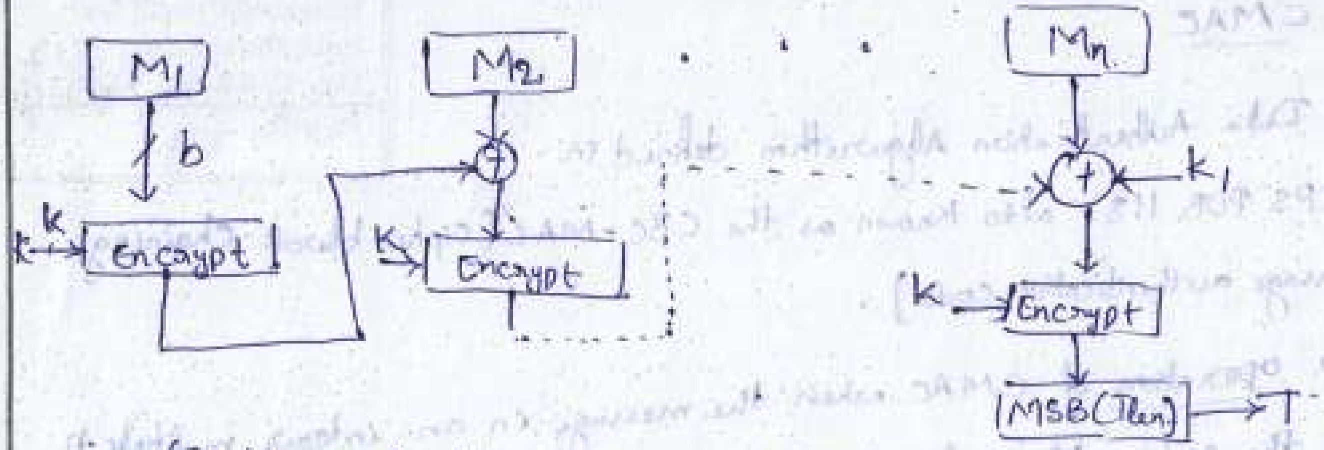
$T = MSB_{Tlen}(C_n)$

where T = message authentication code

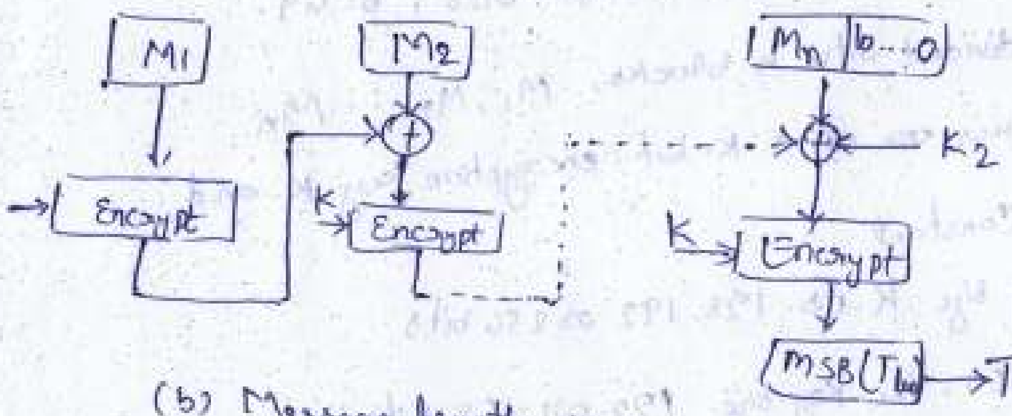
Tlen : bit length of T

$MSB_s(X)$: the s leftmost bits of the bit string X

CMAC



(a) Message length is integer multiple of block size



(b) Message length is not integer multiple of block size

If the message is not an integer multiple of the cipher block length then the final block is padded to the right with a 1 and as many 0's as necessary so that the final block is also of length b. The CMAC operation then proceeds as before, except the different n-bit key K_2 is used instead of K_1

The two n-bit key are derived from the k-bit encryption key as follows

$$L = E(K, 0^n)$$

$$K_1 = L \cdot x$$

$$K_2 = L \cdot x^2 = (L \cdot x) \cdot x$$

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Digital Signatures :-

Unit No:	III
Lecture No:	L40
Link to Session Planner (SP):S.No.	45 of SP
Book Reference:	T1/C13
Date Conducted:	29/9/18
Page No:	65

Requirements :-

Message authentication protect's two parites who Exchange messages from any third party.

However, it does not protect two parties against each other

Requirement's for a digital signature:

→ The signature must be a bit pattern that depends on the message being signed.

→ The signature must use some information unique to the sender, to prevent both forgery and denial.

→ It must be relatively easy to produce the digital signature

→ It must be relatively easy to recognize and verify the digital signature.

→ It must be computationally infeasible to forge a digital signature.

→ It must be practical to retain a copy of the digital signature in storage.

A variety of approaches has been proposed for digital signature function. These approaches fall into two categories:

direct :-

arbitrated :-

Direct Digital Signature:-

The direct digital signature involves only the communicating parties. It is assumed that the destination knows the public key of the source.

A digital signature may be formed by encrypting the entire message with the sender's private key or by encrypting a hash code of the message with sender's private key.

Confidentiality can be provided by further encrypting entire the receiver's public key, or a shared secret key.

Note that it is important to perform the signature function first then an outer confidentiality function.

All direct schemes described so far share a common weakness. The validity of the scheme depends on the security of the sender's private key.

Arbitrated Digital Signature:- The problem with digital signature can be addressed by using arbiter.

Arbitrated Digital signature operates as follows:-

Every signed message from sender X to a receiver Y goes first to an arbiter A, who subjects the message and it's signature to a number of tests to check it's origin and content. The message is dated and send to Y.

The presence of A solves the problem faced by direct signature scheme.

(1) $X \rightarrow A : M \| E(K_{xa}, [ID_x \| H(M)])$

(2) $A \rightarrow Y : E(K_{ya}, [ID_x \| M \| E(K_{xa}, [ID_x \| H(M)]) \| T])$

(a) Conventional Encryption, Arbiter sees message.

(1) $X \rightarrow A : ID_x \| E(K_{xy}, M) \| E(K_{xa}, [ID_x \| H(E(K_{xy}, M))])$

(2) $A \rightarrow Y : E(K_{ay}, [ID_x \| E(K_{xy}, M) \| E(K_{xa}, [ID_x \| H(E(K_{xy}, M))]) \| T])$

b) Conventional Encryption, Arbiter Does not see message.

(1) $X \rightarrow A : ID_x \| E(PR_x, [ID_x \| E(PU_y, E(PR_x, M))])$

(2) $A \rightarrow Y : E(PR_a, [ID_x \| E(PU_y, E(PR_x, M)) \| T])$

(c) Public-key Encryption, Arbiter Does not see message

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Authentication Protocols

Unit No: III
Lecture No: L41
Link to Session Planner (SP): S.No. 46 of SP
Book Reference: T1 / C13
Date Conducted: 29/9/15
Page No: 66

Authentication Protocols can generalise in two areas
- Mutual Authentication
→ one way Authentication

Mutual Authentication :- An important application area is that of mutual Authentication protocols. Such protocols enable communicating parties to satisfy themselves mutually about each other's identity & to exchange session-keys.

Central to the problem of authenticated key exchange are two issues: Confidentiality & timeliness.

→ Symmetric Encryption Approaches :-
A two level hierarchy of symmetric encryption key's can be used to provide confidentiality for communication in a distributed environment. The protocol can be summarized as follows:- [Needham/Schroeder protocol]

① $A \rightarrow KDC : ID_A \| ID_B \| N_1$

② $KDC \rightarrow A : E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$

③ $A \rightarrow B : E(K_b, [K_s \| ID_A])$

④ $B \rightarrow A : E(K_s, N_2)$

⑤ $A \rightarrow B : E(K_s, f(N_2))$

step:1 Secret keys Ka and K_b are shared between A and the KDC and B and the KDC, respectively

step:2 A securely acquires a new session key

step:3 B securely get a session key

step:4 B knowledge the of K_s

step:5 assures B of A's knowledge of K_s & assures B that this is a fresh

Denning proposes to overcome this weakness by a modification to the Needham/Schroeder protocol that includes the addition of a timestamp to steps 2 & 3.

(1) $A \rightarrow KDC : ID_A \| ID_B$

(2) $KDC \rightarrow A : E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$

(3) $A \rightarrow B : E(K_b, [K_s \| ID_A \| T])$

(4) $B \rightarrow A : E(K_s, N_1)$

(5) $A \rightarrow B : E(K_s, f(N_1))$

T is a timestamp that assures A and B that the session key has only just been generated.

The Denning protocol seems to provide an increased degree of security compared to Needham/schroeder protocol.

Improved strategy was presented in [NEUM93A].

1. $A \rightarrow B : ID_A \| N_a$
2. $B \rightarrow KDC : ID_B \| N_b \| E(K_b, [ID_A \| N_a \| T_b])$
3. $KDC \rightarrow A : E(K_a, [ID_B \| N_a \| K_s \| T_b]) \| E(K_b, [ID_A \| K_s \| T_b]) \| N_b$
4. $A \rightarrow B : E(K_b, [ID_A \| K_s \| T_b]) \| K(K_s, N_b)$

Suppose that A and B establish a session using the aforementioned protocol and then conclude that session.

Subsequently, but within the time limit established by the protocol. A desires a new session with B. The following protocol ensure.

(1) $A \rightarrow B : E(K_b, [ID_A \| K_s \| T_b]) \| N'_a$

(2) $B \rightarrow A : N'_b \| E(K_s, N'_a)$

(3) $A \rightarrow B : E(K_s, N'_b)$

When B receives the message in step 1, it verifies that the ticket has not Expired. The newly generated nonces N'_a & N'_b assure each party that there is no replay attack

Class Notes

Subject:
Faculty: ~~P.DEVIKA~~ SK. khaja shareef
Topic: Authentication Protocols-

Unit No : III
Lecture No : L41
Link to Session: 46
Reference to: T_1 | CL3
Planner (SP) :
Date Conducted : 29/9/18
Page No: 67

Public-key Encryption Approaches

one approach to the use of public-key encryption for the purpose of session key distribution.

This protocol assumes that each of the two parties is in possession of the current public key of the other.

A protocol using timestamps is provided

(1) $A \rightarrow AS: ID_A \| ID_B$

(2) $AS \rightarrow A: E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T])$

(3) $A \rightarrow B: E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_s \| T]))$

This Protocol is Compact. but, as before, requires synchronization of clocks. Another approach, proposed by woo and Lam [WOO92a], make use of nonces.

(1) $A \rightarrow KDC: ID_A \| ID_B$

(2) $KDC \rightarrow A: E(PR_{auth}, [ID_B \| PU_b])$

(3) $A \rightarrow B: E(PU_b, [N_a \| ID_A])$

(4) $B \rightarrow KDC: ID_A \| ID_B \| E(PU_{auth}, N_a)$

(5) $KDC \rightarrow B: E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_B]))$

(6) $B \rightarrow A: E(PU_a, E(PR_{auth}, [(N_a \| K_s \| ID_B) \| N_b]))$

(7) $A \rightarrow B: E(K_s, N_b)$

This seems to be a secure protocol that takes into account the various attacks. but authors spotted a flaw & submitted a revised version of the algorithm.

① $A \rightarrow KDC : ID_A \| ID_B$

② $KDC \rightarrow A : E(PR_{auth}, [ID_B \| PU_b])$

③ $A \rightarrow B : E(PU_b, [N_a \| ID_A])$

④ $B \rightarrow KDC : ID_A \| ID_B \| E(PU_{auth}, N_a)$

⑤ $KDC \rightarrow B : E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B]))$

⑥ $B \rightarrow A : E(PU_a, E(PR_{auth}, [(N_a \| K_s \| ID_A \| ID_B) \| N_b]))$

⑦ $A \rightarrow B : E(K_s, N_b)$

One-way Authentication :-

One application for which encryption is growing in popularity is electronic mail (e-mail).

The very nature of electronic mail, and its chief benefit, is that it is not necessary for the sender and receiver to be online at the same time.

A second requirement is that of authentication.

Symmetric Encryption Approach :-

Using symmetric encryption, the decentralized key distributed scenario illustrated is impractical.

This scheme requires the sender to issue a request to the intended recipient, a wait a response that includes a session key, and only then send the message.

① $A \rightarrow KDC : ID_A \| ID_B \| N_1$

② $KDC \rightarrow A : E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$

③ $A \rightarrow B : E(K_b, [K_s \| ID_A]) \| E(K_s, M)$

UNIT NO: III
Lecture NO: 41
Reference : T_1/C_{13}
Link to session:- 46
Date : 29/9/18
Page NO:- 68

Topic :- Authentication Protocols.

One way Authentication :

Public-key Encryption Approaches

If Confidentiality is the primary Concerns.

$$A \rightarrow B : E(PU_b, k_s) \| E(k_s, M)$$

If authentication is the primary Concern.

$$A \rightarrow B : M \| E(PR_a, H(M))$$

To the message & signature can be encrypted

$$A \rightarrow B : E(PU_b, [M \| E(PR_a, H(M))])$$

$$A \rightarrow B : M \| E(PR_a, H(M)) \| E(PR_{as}, [T \| ID_A \| PU_a])$$

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: Digital Signature standard

Unit No: III
Lecture No: L42
Link to Session
Planner (SP):S.No 47 of SP
Book Reference: T1 / C13
Date Conducted: 1/11/18
Page No: 69

The Digital Signature standard (DSS) was adopted by NIST published DSS as Federal Information processing standard FIPS186.

DSS uses a digital signature Algorithm (DSA) based on the ElGamal scheme

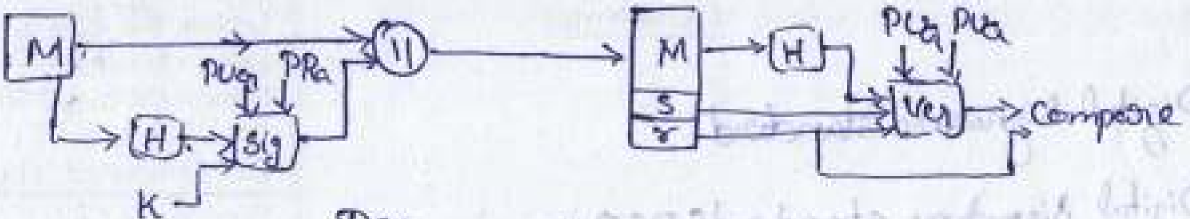
Dss has been Criticized from the time it was published the main complaint regards the secrecy of DSS design. second complaint is size of prime, 512 bits

Later NIST made the size variable to respond to this complaint

The DSS Approach -

The DSS uses an algorithm that is designed to provide only the digital signature function.

The DSS approach also makes use of a hash function. the hash code is provided as input to a signature function along with a random number k generated for this particular signature. The signature function along with a random number k generated also depends on the sender's private key (PR_a) and set of parameters known to a group of communicating Principals. Consider this set of parameters known a global Public key (PU_G). The result is a gt signature consisting two components, labeled s and r.



DSS approach

At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function. also the verification function also depends on the global public key as well as the sender's public key.

The output of verification function is a value that is such that only the sender, with knowledge of the private key, could have produced the valid signature.

The Digital Signature Algorithm.

The DSA is based on the difficulty of computing discrete algorithms and is based on schemes originally presented by ElGamal and Schnorr.

Global Public key Components.

→ Select a prime number p where $2^{L-1} < p < 2^{L}$ for $512 \leq L \leq 1024$ and L a multiple of 64 :-

→ Select a q prime number of a 160 bits such q prime devisor of $(p-1)$ where $2^{159} < q < 2^{160}$

→ $g = h^{(p-1)/q} \bmod p$, where h is any integer with $1 < h < (p-1)$ such that $h^{(p-1)/q} \bmod p > 1$.

User's Private Key

x is pseudorandom integer with $0 < x < q$.

User's Public key

$y = g^{x} \bmod p$.

Subject: Cryptography and Network Security **Class Notes**
Faculty: Sk.Khaja Shareef
Topic: DSS-2

Unit No:	III
Lecture No:	L42
Link to Session Planner (SP):S.No.	47 of SP
Book Reference:	T1, C13
Date Conducted:	1/10/18
Page No:	30

→ User's Per-Message Secret Number

k = random or pseudorandom integer with $0 < k < q$

→ Signing

$$r = (g^k \bmod p) \bmod q$$

$$s = [k^{-1}(H(M) + xr)] \bmod q$$

Signature = (r, s)

→ Verifying

$$w = (s')^{-1} \bmod q$$

$$u_1 = [H(M')w] \bmod q$$

$$v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$$

→ Test: $v = r'$

- M → message to be signed, $H(M)$ = hash of M using SHA-1

M', r', s' = received version of M, r, s.

→ With these values in hand, user selects a private key and generate a public key.

→ The Private key x must be a number from 1 to $(q-1)$

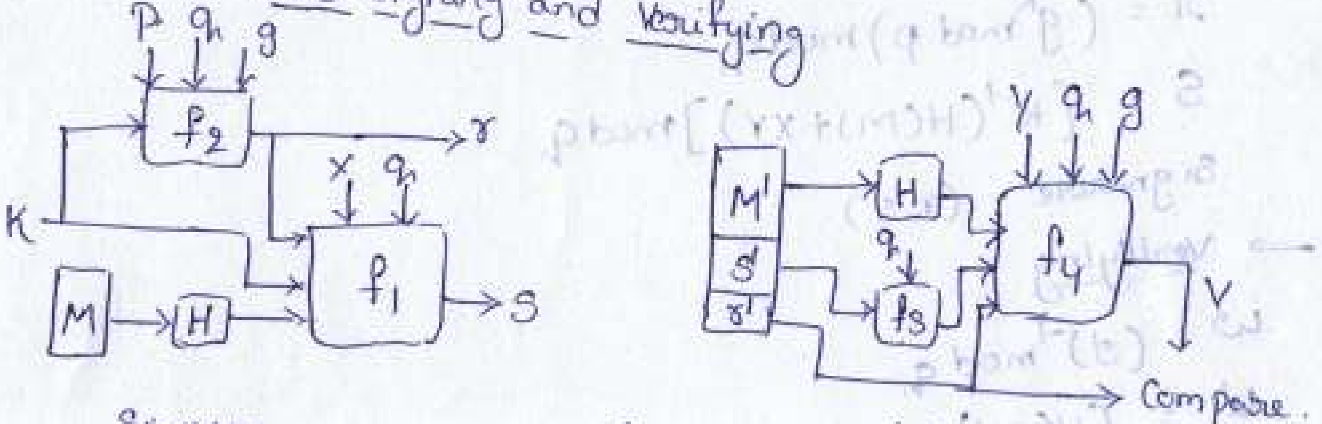
→ The Public key is calculated from private key as $y = g^x \bmod p$.

→ for given y it is relatively infeasible to find x

To create a signature, a user calculate two quantities, r and s that are functions of the public key components (p, q, g), the user private key (x), the hash code of the message, $H(m)$, and additional integer k that should be generated randomly.

At the receiving end, verification is performed using the formulas. The receiver generates a quantity V that is function of the Public key component, the sender's public key and the hash code of the incoming message. If this quantity matches the r component of the signature, then the signature is validated.

DSS signing and verifying



Signing

1. Alice choose a random number r $(1<r\leq q)$
2. Alice calculate the first signature
   $r_1 = (g^k \bmod p) \bmod q$.
   Note value of the first signature does not depend on M
3. Alice creates a digest of message $h(m)$
4. Alice calculate second signature
   $S = (K^{-1}(H(m)+rx)) \bmod q$
5. Alice sends M, r, S to bob.

Verifying

1. Bob checks to see if $0 < S_1 < q$
2. Bob checks to see if $0 < r < q$
3. Bob calculates a digest of M using the same hash algorithm using by Alice.
4. Bob calculate V

$$w = f_3(s', q) = (s)^{-1} \bmod q$$

$$V = f_4(y, q, g, H(m), w, r')$$
$$= ((g^{(H(m)w)} \bmod q \; y^{rw} \bmod q) \bmod p) \bmod q$$

If r is congruent to V, the message is accepted. otherwise rejected