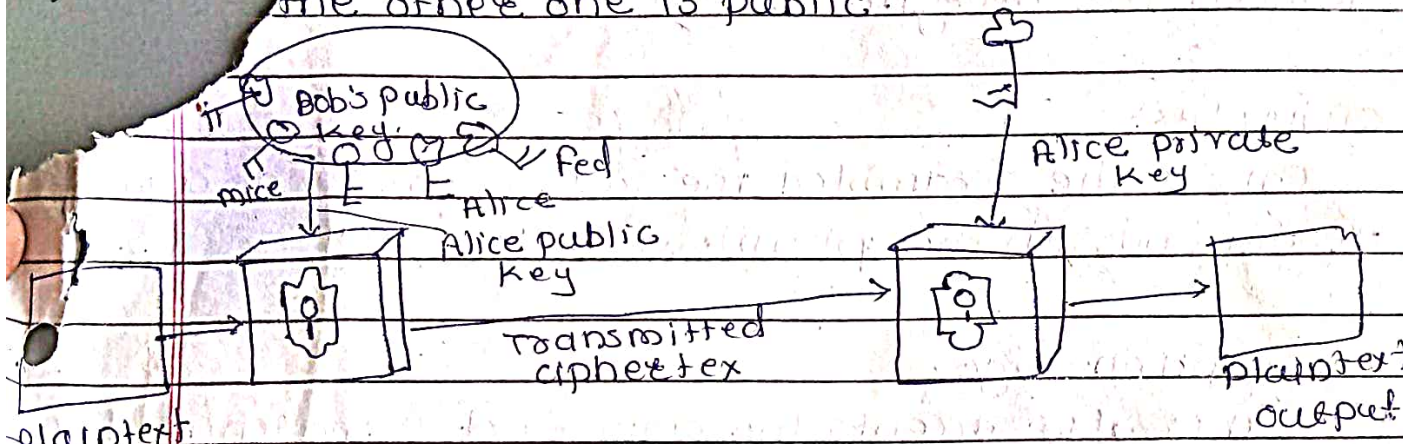
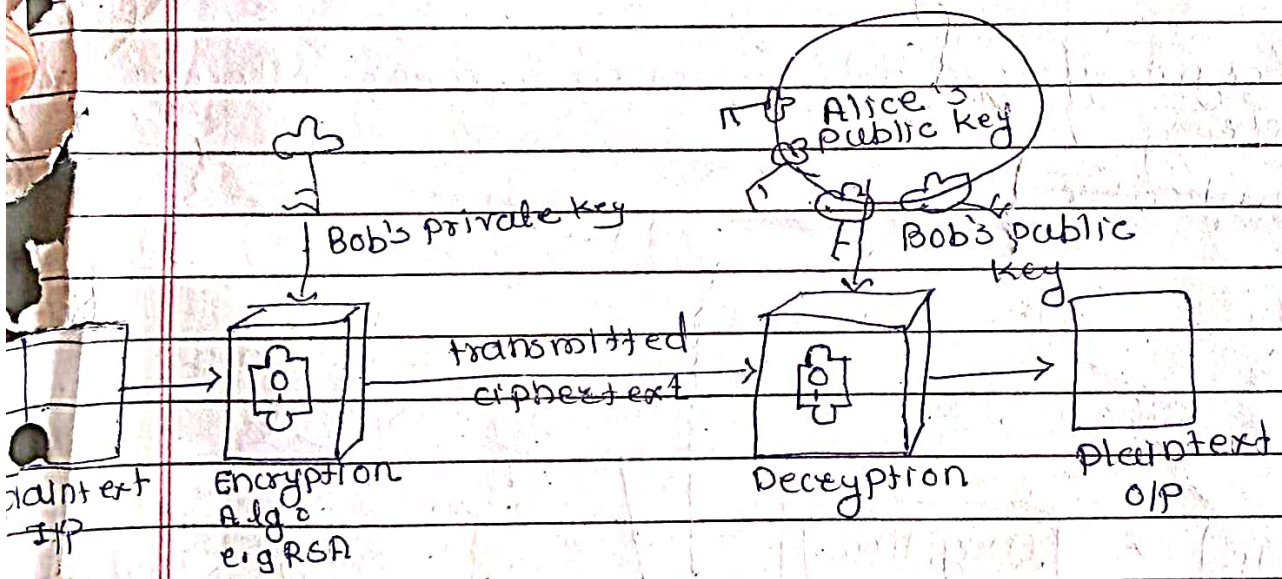
Public key Cryptosystem, Key Management & Authentication

Principles of Public Key Cryptosystem

ic key cryptography is a cryptographic system
t uses two separate keys, one of which is secrete
the other one is public.



a) Encryption



b) Authentication

A public key encryption scheme has 6 parts.

1) Plaintext
This is the readable message or data that is fed into the algorithm as input.

2) Encryption algorithm.
The encryption algo. performs various transformations on the plaintext.

3) Public & Private key

This is a pair of keys that have be
so that if one is used for encryption,
is used for decryption.

4) ciphertext

This is the scrambled message produced as output
It depends on the plaintext & the key.

5) Decryption algo.

This algo accepts the ciphertext & matching key &
produces the original plaintext.

Any cryptosystem are designed to meet following
1) secrecy
2) Authentication

Public key cryptosystem : secrecy

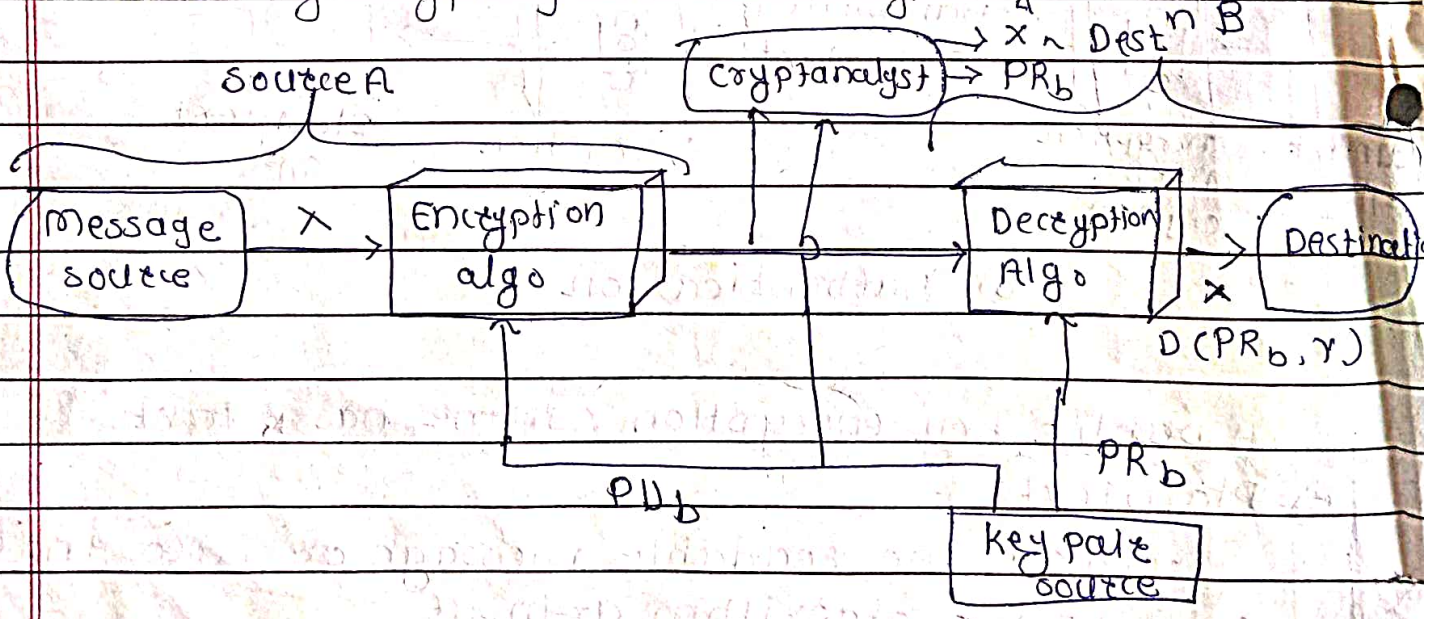


fig - encryption using public key cryptography

- Suppose there is some source A that produces a msg in plaintext $X = [X_1, X_2 \cdots X_m]$ & sends it to B
- B generates a related pair of keys: a public key PU_b & private key PR_b. PU_b is publicly available & therefore accessible by A.
- With the msg. X & the encryption key PU_b as input A forms the ciphertext $Y = [Y_1, Y_2 \cdots Y_N]$

$$Y = E(PU_b, X)$$

- The intended receiver, having the matching private key, is able to decrypt the msg.

$$X = D(PR_b, Y)$$

- An adversary, observing Y and having access to PU_b only & may attemp to recover X & or PR_b
- If the adversary is interested only in this particular msg. then the focus of effort is to recover X by generating a plaintext estimate.
- whereas if the adversary is interested in being able to read future msg. as well as then the attempt to recover PR_b by generating an estimate PR_b.

Public Key encryption & authentication

- However, the diagram does not provide authentication of sender as anyone having access to the public key can encrypt the msg.
- Public key encryption can be used to provide authentication in the foll. manner.

- When A wishes to send msg. to B where confidentiality is not needed but authentication is required, A encrypts the msg. using PR_a

- Anyone having access to PU_a can decrypt the msg. However one thing is sure that the msg. originated from A since no one except A could have encrypted the message using PR_a.

- A prepares a msg. to B & encrypts it using A's private key before transmitting it

$$Y = E(PR_a, X)$$

- B can decrypt msg using A's public key.

$$X = D(PU_a, Y)$$

- Because the msg. was encrypted using A's private key, only A could have prepared the msg.
- In addition it is impossible to alter the msg. without access to A's private key, so the msg. is authenticated.
- It must have the property that it is infeasible to change the document without changing the authenticator.
- If the authenticator is encrypted with the sender's private key, it serves as a signature

Page No.
Date

Public key cryptosystem : Authⁿ & secrecy

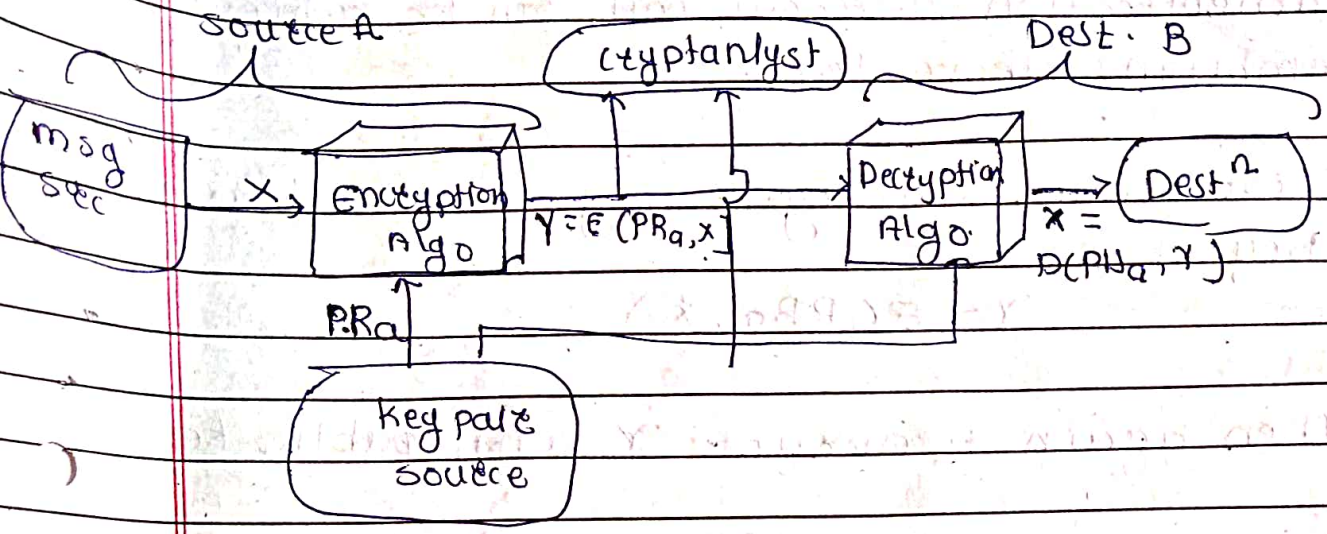
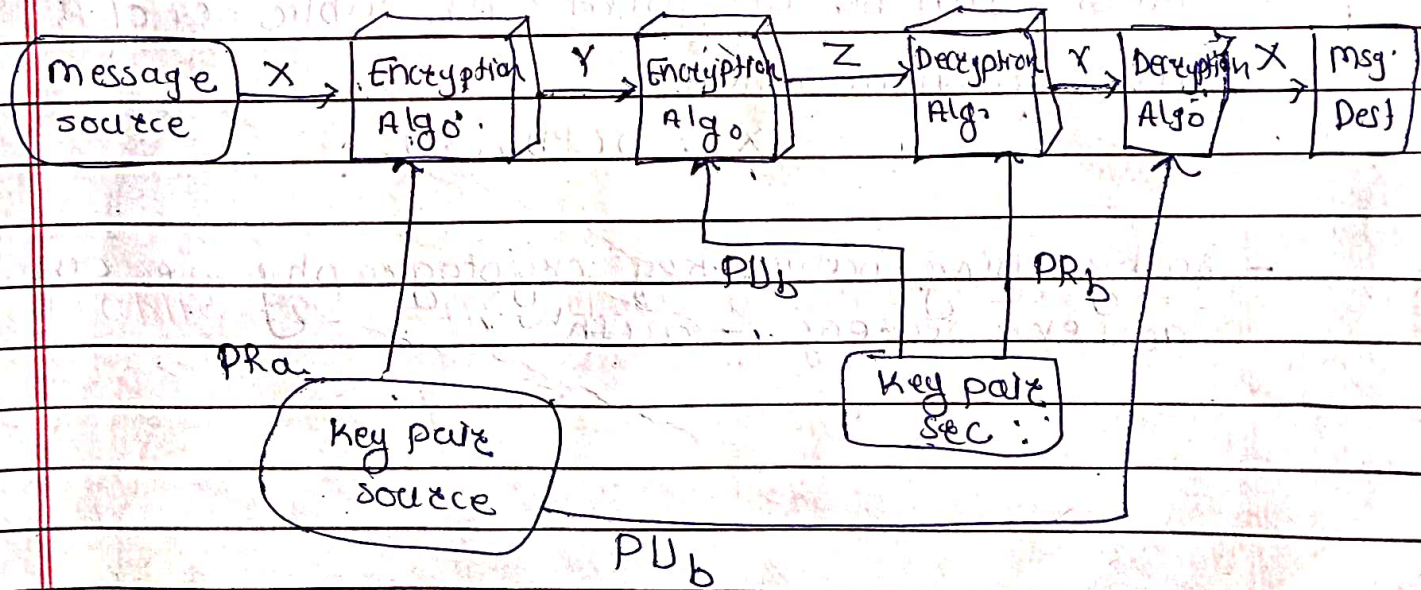


fig. Authentication using public key cryptography

Public key cryptosystem : Authentication & secrecy.



Secrecy → PUb PRb
Authentication PRa PUa

Authentication & secrecy both can be achieved by combining above tech.

- First sender A encrypt msg: X with private key of A

$$Y = E(PR_a, X)$$

- Then again A encrypt Y with public key of B

$$Z = E(PU_b, Y)$$

- Then send Z
- only B can decrypt Z as it is encrypted with public key of B, so it gives secrecy,

$$Y = D(PR_b, Z)$$

- Now Y can be decrypted with public key of A; so it gives authentication

$$X = D(PU_a, Y)$$

- so by using public key cryptography we can achieve secrecy & authn.

Lucky | Page No.:
Date : / /

RSA Algorithm

→

- It is a merde from initial letters of the 3 scientist Rivest, Shamir & Adleman developed in 1978
- 2 keys are used that are public & private key
- Public key user A is used for encryption we have to use private key of same user for decryption.
- RSA is a block cipher in which the plaintext & cipher text are integers betn of the 0-1 for same value n
- RSA algorithm processes plaintext blocks, with each block having a binary value less than some no. n
- The block size must be less than or equal to $\log_2(n)+1$

Algorithm

i) select two large numbers p & q

ii) calculate $n = pq$

iii) calculate $\phi(n) = (p-1) * (q-1)$

iv) choose value of e

$1 < e < \phi(n)$ & $\gcd(\phi(n), e) = 1$

v) compute d such that $d * e = 1 \pmod{\phi(n)}$

- RSA is a public key algorithm with public key $PU = \{e, n\}$ & private key $PR = \{d, n\}$
- Encryption & decryption are of the following from, for some plaintext block m & ciphertext c

$$c = m^e \bmod n$$

$$m = c^e \bmod n$$

$$m = (m^e)^{e \bmod n}$$

- For the above equation to be true, d must be an inverse of e
- D can be calculated from e using extended Euclids algorithm.
- Both sender & receiver must know the value of n
- The sender knows the value of e & only the receiver knows the value of d.
- RSA can also be subjected to various attacks like brute force attack, various mathematical attack. timing attack & chosen ciphertext attack.
- Some of these attacks exploit the mathematical characteristics of RSA.

Example

1) a) $p = 17$ & $q = 11$.
b) $n = pq = 17 \times 11 = 187$
c) $\phi(n) = (p-1) * (q-1) = 16 \times 10 = 160$
d) Let e be the 7
e) $d = e^{-1} \bmod 160 = 23$
f) Now, PU = {7, 187} & PR {23, 187}
if $m = 88$ then by RSA

Encryption

$$C = 88^7 \bmod 187$$
$$= [88 \times 88^2 \times 88^4] \bmod 87$$

Decryption

Here $C = 11$
$$m = 11^{23} \bmod 187$$
$$= [11 \times 11^2 \times 11^4 \times 11^5 \times 11^3] \bmod 187$$
$p = 4 \quad q = 9$

RSA algorithm example (asymmetric)

1) Two prime no.
   $P = 3$, $q = 5$

2) $n = P \times q = 3 \times 5 = 15$

   $\boxed{n = 15}$

3) $\phi(n) = (P-1)(q-1) = 2 \times 4$
   $= 8$

4) Assume e such that $\gcd(e, \phi(n)) = 1$ &
   $1 < e < \phi(n)$

   $\gcd(3, 8) = 1$ ✓ we take this
   $\gcd(5, 8) = 1$
   $\gcd(7, 8) = 1$ $\boxed{e = 3}$

5) Find d
   $d \times e \bmod \phi(n) = 1$ — $d \times 3 \bmod 8 = 1$
   $d \times 3 \bmod 8 = 1$ $d \times ? = \frac{1}{2}$
   consider $d = 3$ ∴

   $3 \times 3 \bmod 8 = 1$
   $9 \bmod 8 = 1$
   $1 = 1$

public key = $\{e, n\} = \{3, 15\}$
private key = $\{d, n\} = \{3, 15\}$

Date:

Encryption

$e = 5 \Rightarrow d$

consider plaintext = P = 8

$C = P^8 \bmod n$

$= 8^3 \bmod n$

$= 512 \bmod 15$

$\boxed{C = 2}$

Decryption

$P = C^d \bmod n$

$P = 2^3 \bmod 15$

$P = 8 \bmod 15$

$\boxed{P = 8}$

(2) i) $P = 13$, $q = 11$

ii) $n = P \times q = 143$

iii) $\phi(n) = (P-1) \times (q-1) = 12 * 10 = 120$

iv) $13 = e$

v) $d = e^{-1} \bmod \phi(n)$ or ~~$ed = 1 \bmod \phi(n)$~~

~~$13 \times d \bmod 120 = 1$~~

$d = ((\phi(n) * i) + 1) / e$

$d = (120+1)/13 = 9.80$ ($\because i = 1$)

$d = (240+1)/13 = 18.53$ ($\because i = 2$)

$d = (360+1)/13 = 27.76$ ($\because i = 3$)

$d = (480+1)/13 = 37$ ($\because i = 4$)

$\boxed{d = 37}$

vi) $P_U(e, n)$ & $P_E(d, n)$
$(13, 143)$ & $(37, 143)$

vii) $C = P^e \bmod n$
$P = 13$ where $P < n$
$C = P^e \bmod n = 13^{13} \bmod 143 = 52$

$\boxed{C = 52}$

viii) d $P = C^e \bmod n = 52^{37} \bmod 143 = 13$

$\boxed{P = 13}$

* Exercise

1) P & Q are two $P = 7$ & $Q = 17$, Take public key $E = 5$, if plaintext value is 6 then what will be cipher text value according to RSA algo. & Again calculate plain text value from cipher text.

2. In public key cryptosystem using RSA algo. User uses two prime no. 5 & 7. He chooses 11 as encryption key, find out decryption key, what will be cipher text, if plaintext is 2 & Decrypt cipher text, what will be the value of plain text &

3. Two prime no. $P = 13$ & $Q = 17$, Take public key $E = 19$, if original message is 12, then what will be cipher text value & private key value according to RSA algo & Again calculate plaintext value from cipher text

Page No
Date:

④ $p = 5$, $q = 11$, $e = 3$ & $m = 9$

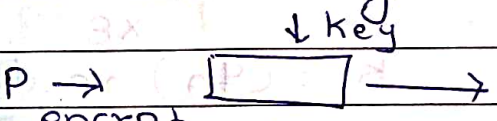
5) $p = 11$, $q = 3$, $e = 11$, $m = 7$

6) $c = 14$ $e = 7$ $n = 33$

* Diffie - Hellman key exchange

- It is an not encryption algorithm means in that plaintext, ciphertext, decryption are not included.
- It is used to exchange secrete keys between two users
- We will use asymmetric encryption to exchange the secrete key between the users

→ Public & Private



Attacker does not access the key.

- why use diffie Hellman ?
  - because when we are sending a key to receiver, it can be attacked in between.

Algorithm

→ i) consider a prime no. 'q'
ii) select α such that it must be the primitive root of q & $\alpha < q$

Primitive root

'a' is a primitive root of q is

a mod q
a^2 mod q
a^3 mod q
a^{q-1} mod q

} extend power when we

e.g q = 7
check 3

$3^1 \bmod 7 = 3$
$3^2 \bmod 7 = 2$
$3^3 \bmod 7 = 6$
$3^4 \bmod 7 = 4$
$3^5 \bmod 7 = 5$
$3^6 \bmod 7 = 1$

iii) Assume X_A (Private key) ($X_A < q$) & User A

iv) Calculate Y_A (Public $Y_A = \alpha^{X_A} \bmod q$

iv) Assume X_B (Private Key of User B) $X_B < q$

Calculate Y_B (Public key of User B) $Y_B = \alpha^{X_B} \bmod q$

v) Key generation

A User sender	User B receive r
$K = (Y_B)^{X_A} \bmod q$	$K = (Y_A)^{X_B} \bmod q$

E

i) $q = 11$

ii) $\alpha =$

$\alpha^1 \bmod 11$
$\alpha^2 \bmod 11$
$\alpha^3 \bmod 11$
$\vdots$
$\alpha^{10} \bmod 11$

$= \{1, 2, 3 \ldots\ldots 10\}$

vi) Calculation of secret key by User A

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

vii) Calculation of secret key by User B

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

Steps

1) Assume prime No. q
2) Select α such that
   i) α is primitive root of q
   ii) $\alpha < q$
3) Assume
   X_A as private key of User A
   $X_A < q$

   calculate $Y_A \rightarrow$ Public key of User A
   $Y_A = \alpha^{X_A} \bmod q$
   $\{ \overset{Pr}{X_A}, \overset{Pu}{Y_A} \}$
   → User A's key
4) Assume $X_B \rightarrow$ Private key of User B
   $X_B < q$

   calculate $Y_B \rightarrow$ Public key of User B
   $Y_B = \alpha^{X_B} \bmod q$
   $\{X_B, Y_B\}$ User B's key
5) Key generation

User A	User B
$K = (Y_B)^{X_A} \bmod q$	$K = (Y_A)^{X_B} \bmod q$

Example (1) $q = 17$ $\quad \alpha = 5$
A' secrete key (Private) 4
B' secrete key (Private) 6

(2) $q = 7 \quad \alpha = 3$
A Pr → 2
B → 5

Page No.:
Date:

Ex →

i) $q = 11$

ii) $\alpha = 2$ < primitive root of q, $\alpha < q$

$\alpha^1 \bmod 11 \quad \alpha^2 \bmod 11 \quad \ldots\ldots \quad \alpha^{10} \bmod 11 =$

$\{1, 2, 3, \ldots 10\}$

power → / Number ↓	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	5	10	9	7	3	6	1
3										
4										
5										
6										
7										
8										
9										
10										
11										

iii) select $X_A < q$

$X_A = 8$

$Y_A = \alpha^{X_A} \bmod q = 2^8 \bmod 11 = 3$

iv) select $X_B = 4$

$Y_B = \alpha^{X_B} \bmod q = 2^4 \bmod 11 = 5$

so user A = $\{X_A = 8, Y_A = 3\}$

user B = $\{X_B = 4, Y_B = 5\}$

v) key generation

user A (sender)

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

$= 5^8 \bmod 11$

$K = 4$

user B (Receiver)

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

$= 3^4 \bmod 11$

$K = 4$

Page No.:
Date:

Ex →

i) $q = 11$

ii) $\alpha = 2$ ← primitive root of q, $\alpha < q$

$\alpha^1 \bmod 11 \quad \alpha^2 \bmod 11 \quad \ldots\ldots \quad \alpha^{10} \bmod 11 =$

$\{1, 2, 3, \ldots 10\}$

power → / Number ↓	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	5	10	9	7	3	6	1
3										
4										
5										
6										
7										
8										
9										
10										
11										

iii) select $X_A < q$

$X_A = 8$

$Y_A = \alpha^{X_A} \bmod q = 2^8 \bmod 11 = 3$

iv) select $X_B = 4$

$Y_B = \alpha^{X_B} \bmod q = 2^4 \bmod 11 = 5$

so user A = $\{X_A = 8, Y_A = 3\}$

user B = $\{X_B = 4, Y_B = 5\}$

v) Key generation

user A (sender)

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

$= 5^8 \bmod 11$

$K = 4$

user B (Receiver)

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

$= 3^4 \bmod 11$

$K = 4$

i) Msg Encryp
ii) MAC → fixed size block of data (compress)
iii) Hash fun.

MAC

* Message Authentication Code (generate MAC & append with)

- An cryptographic checksum or MAC is a function of the message & secrete key that produces a fixed length value that serves as the authenticator.
- This technique assumes that two communicating parties say A & B share a common secrete key K, when A has message to send to B, it calculate the MAC as function of the msg & key

MAC = C(K, M)
↙ sender & Receiver

where,

M = input message
C = MAC function
K = shared secret key
MAC = Message authentication code

- The msg plus MAC are transmitted to the intended recipient
- The recipient performs the same calculation on the received msg. using the same secret key to generate a new MAC.
- The received MAC is compared to the calculated MAC.

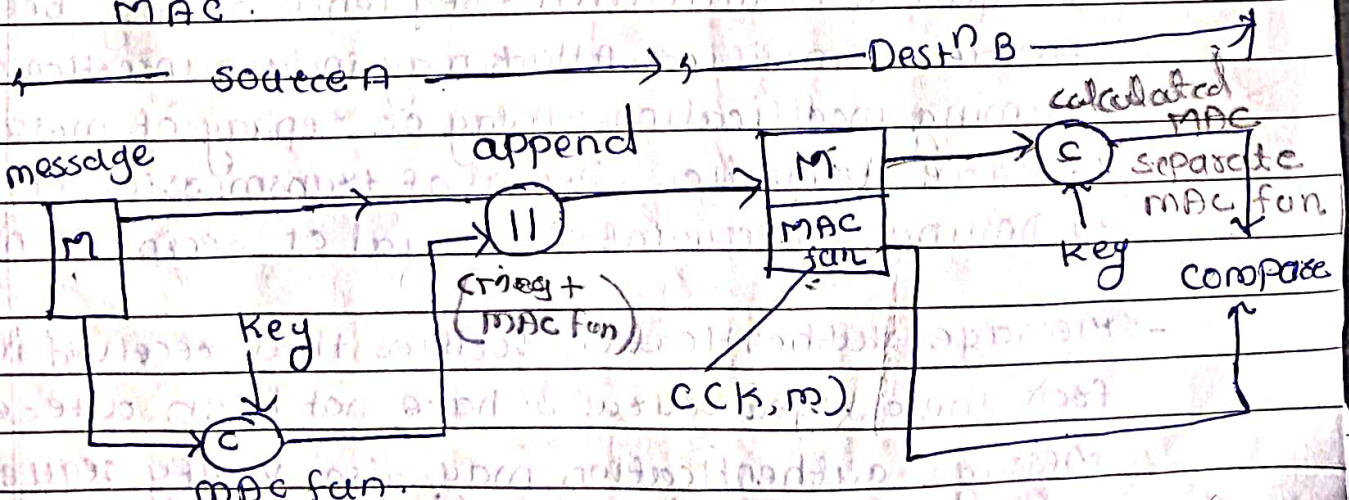


fig → Authentication using MAC no confidentiality

separate MAC fun & compare with the original MAC

* Authentication Requirements

- Message authentication is a mechanism or service used to verify the integrity of a message
- Message authentication assures that data received are exactly as sent by (i.e. contain no modification, insertion, deletion or replay) & that the purported identity of the sender is valid.

Need for msg authentication

Following attacks are possible which are the reason why authentication is needed.

1) Disclosure

Release of msg contents to any person not knowing the secret key.

2) Traffic analysis -

Discovery of the pattern of traffic betn parties. Traffic analysis reveals info. like the freequency & length of msg betn parties & communicating parties could be determined.

3) masquerade - Impersonating other person & sending msg.

4) Content modification - changes are made to the contents of msg. changes may include insertion, deletion modification

5) sequence Modification - sequence of msg between parties is modified. Attack may include insertion deletion & reordering

6) Timing modification - Delay or replay of message

7) source repudiation - Denial of transmission of msg by source

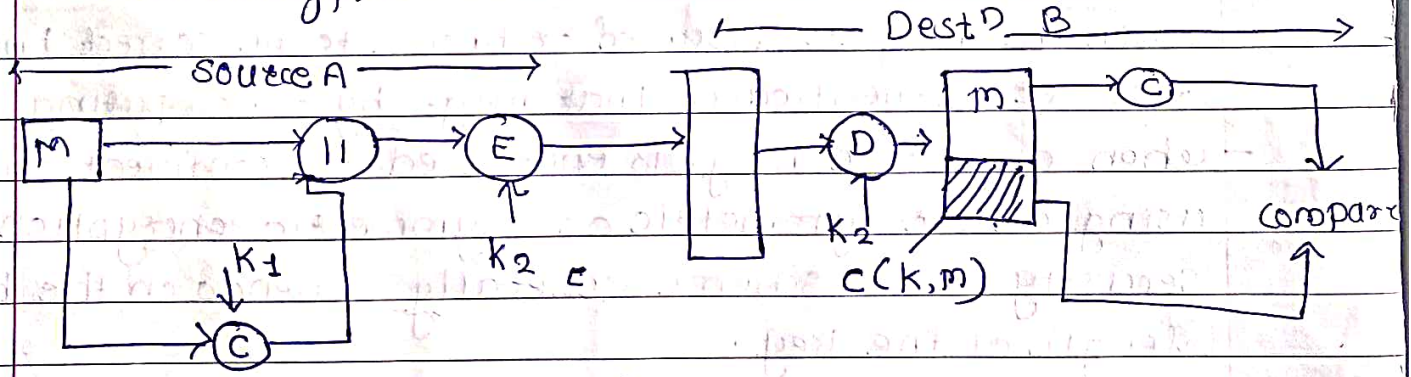
8) Destination repudiation : Denial of recipt of msg by dest^n

- Message authentication verifies that received msg come from the alleged source & have not been altered.
- Message authentication may also verify sequencing & timeliness.

- Since only the receiver & the sender know the secrete key & if the received MAC matches the calculated MAC then.

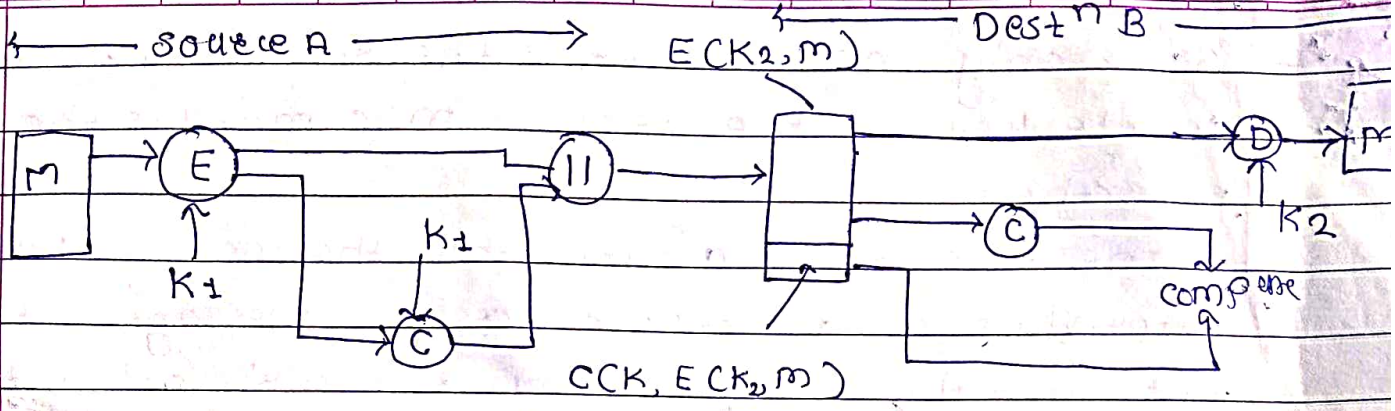
The receiver is assured that the message has not altered. If an attacker alters the message but does not alter the MAC then the receiver's calculations of the MAC will differ from the received MAC. The receiver is assured that the message is from the alleged sender. Because no one else knows the secret key.

- confidentiality can be provided by performing msg encryption either after or before the MAC algorithm
- In both cases, two separate keys are needed, each of which is shared by the sender & receiver.
- MAC can be calculated with the message as input then concatenated to the message. The entire block is then encrypted.



Authentication & confidentiality using MAC

- It is preferable to tie the authentication directly to the plaintext, hence the above method is typically preferred.
- Alternately, msg is encrypted 1st then the MAC is calculated using the resulting cipher text & is concatenated to the cipher text



- Authentication & confidentiality using MAC

Requirements for MAC

- A MAC also known as a cryptographic checksum, is generated by a fun: C of the form

$$T = MAC(K, m)$$

where

m is a variable length message

K is a secret key shared only by sender & receiver.

MAC(K, m) is the fixed length authenticator also called tag

- The tag is appended to the msg at the source at a time when the msg is assumed or known to be correct the receiver authenticates that msg by re-computing tag
- When an entire msg is encrypted for confidentiality using either symmetric or asymmetric encryption, the security of the scheme generally depends on the bit length of the key.
- Barring some weakness in the algo, the opponent must resort to brute force attack using all possible keys.
- On average, such an attack will require 2^{K-1} attempts for K-bit key. In particular, for a cipher text only attack, the opponent, given cipher text C performs $P_i = D(K_p, C)$ for all possible key values K_i until a p_i is produced that matches the form of acceptable plaintext

- Then the MAC fun. should satisfy the foll. requirements
1) If an opponent observes m & MAC(K,m) it should be computationally infeasible for the opponent to construct msg m such that MAC(K,m) = MAC(K,m')
2) MAC(K,m) should be uniformly distributed in the sense that for randomly chosen msg, m & m', the probability that MAC(K,m) = MAC(K,m') is 2^{-n} where n is the no. of bits in the tag.
3) Let m' be equal to some known transformation on m. That is m' = f(m), for example, f may involve inverting one/more specific bits. In that case Pr[MAC(K,m) = MAC(K,m')] = 2^{-n}.

Security of MAC / Attack on MACs

We group attacks on MACs into two categories
i) Brute Force Attacks
ii) Cryptanalysis.

1) Brute-Force Attacks
- A brute force attacks on MAC requires more than known msg. MAC pairs than a brute force attack on a has fun.
- There are two types of possible attacks.
i) attack the key space
ii) attack the MAC value

i) Attacking the key space
- If an attacker can determine the MAC key, then it is possible to generate a valid MAC value for any input.
- suppose the key size is k bits & that the attacker has one known text tag (MAC) pair.
- The attacker can then compute the n-bit tag on the known text for all possible keys.

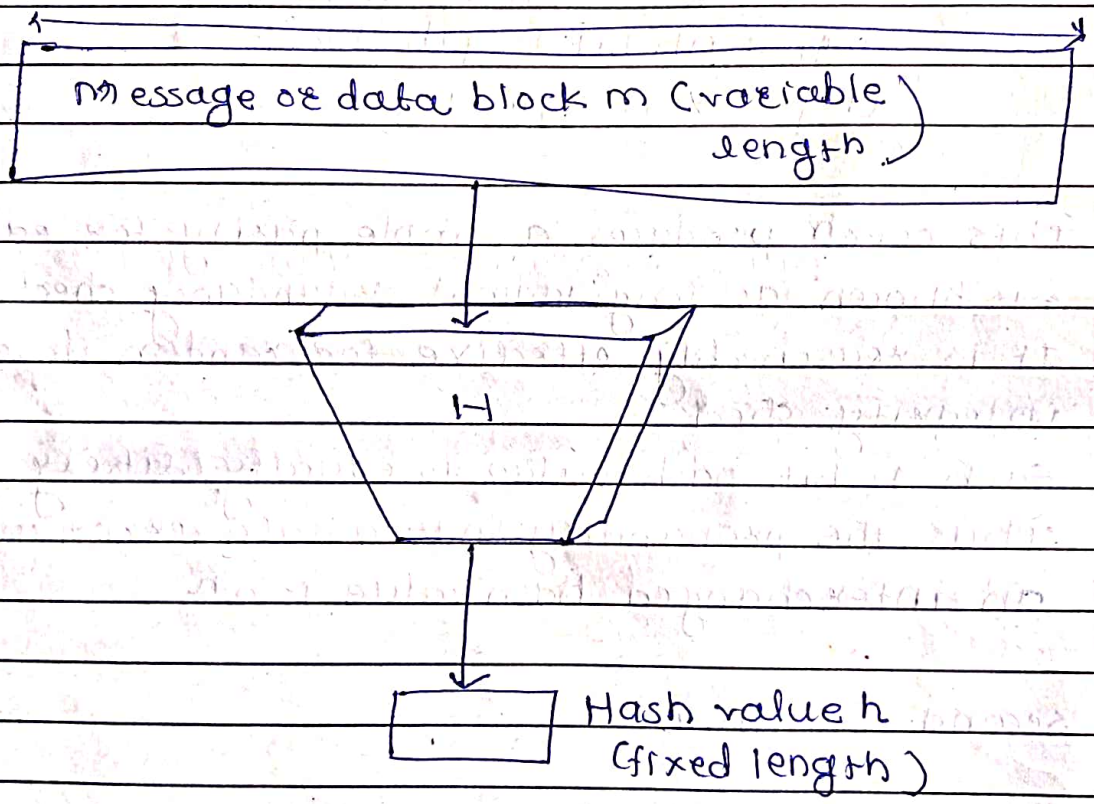
- At least one key will produce the correct MAC value for the msg. Till Now the level of effort is 2^K.
- However, the MAC is a many-to one mapping, so there may be other keys that produce the correct value.
- Thus, if more than one key is found to produce the correct value, additional text-tag pairs must be tested.
- The level of effort becomes less with each additional text MAC pair & after 2 or 3 levels a single key is obtained.

2) Attacking the MAC value
- The attacker will try to generate a valid MAC for a given msg or to find msg. that matches given MAC value.
- Here the value of effort is that of 2^n.
- This attack can't be conducted off line without further input, the attacker will require chosen text tag pairs or knowledge of the key.

2) Cryptanalysis
- cryptanalysis attacks on MAC algo. try to exploit some property of the algo. to perform some attack other than an exhaustive search.
- The way to measure the resistance of MAC algo. to cryptanalysis is to compare it's strength to the effort required for a brute force attack.
- An ideal MAC algo. will require a cryptanalytic effort greater than or equal to the brute force effort.

* Hash function

- A has function H accepts a variable length block of data as input & produces a fixed size hash value.
- A good hash fun. has the property that the results of applying the fun. to a large set of inputs will produce outputs that are evenly distributed & apparently random.



- Two simple, insecure hash fun. are shown here.
- All hash fun. operate using the foll. principles.
  - The input (msg, file etc) is viewed as sequence of n bit blocks.
  - The input is processed one block at a time in an iterative fashion to produce a bit hash fun.

1) First function

- One of the simplest hash fun. is the bit-by-bit exclusive OR (XOR) of every block.
- This can be expressed as

$$C_i = b_{i1} \oplus b_{i2} \oplus \cdots \oplus b_{im}$$

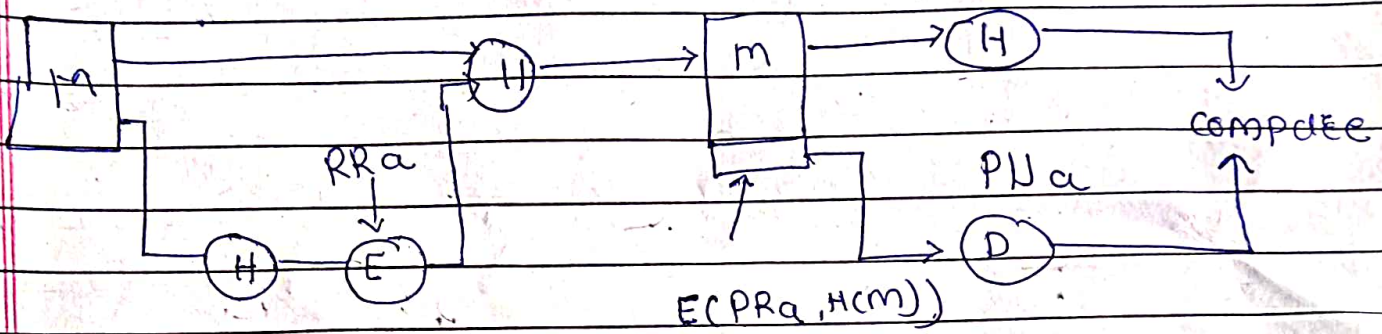
where

C_i = ith bit of the hash code
m = no. of m bit blocks
b_{ij} = ith bit in jth block
$\oplus$ = XOR operⁿ

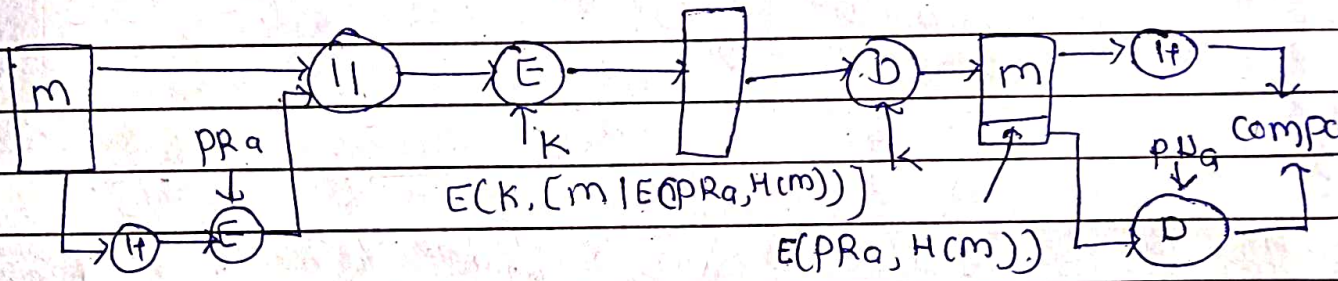
- This operⁿ produces a simple parity for each bit p[illegible] & is known as longitudinal redundancy check.
- It is reasonably effective for random data as dat[illegible] integrity check.
- Each n bit hash value is equally likely
- Thus, the probability that a data error will result in an interchanged hash value is 2^n.

2) Second function

-
- A simple way to improve matters is to perform a one bit circular shift or rotation on the hash value after each block is processed.
- The procedure can be summarized as follows
  1) Initially set the n-bit hash value to zero.
  2) Process each successive n-bit block of data as follows
     a) Rotate the current hash value of the ~~data as follows~~ left by one bit
     b) XOR the block into the hash value



(a)



(b)

a) The hash code is encrypted using public key encryption with senders private key.
This provides authentication. It also provides a digital signature because only the sender could have produced the encrypted hash code.

b) If confidentiality ~~hash fun~~ as well as digital signature is desired, then the msg. plus the private key encrypted hash code can be encrypted using symmetric secret key.

Page No.
Date

* Key Management

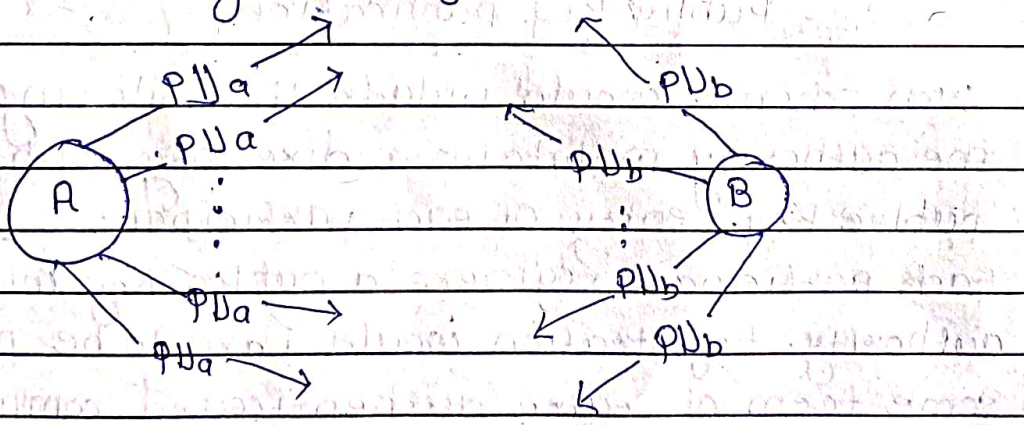
There are two types of public key cryptography.
- The distribution of public keys
- The use of public key encryption to distribute secrete key

Distribution of Public key

i) Public announcement
ii) Publicly available directory
iii) Public key authority
iv) Public key certificates.

i) Public announcement

- If there is some broadly accepted public-key algorithm such as RSA, any participant can broadcast key to the community at large.



- Although this approach is convenient, it has a major weakness
- Some user could pretend to be user A & send a public key to another participant or broadcast such a public key.
- Until such time as user A discovers the forgery & alerts other participants, the forger is able to read all encrypted msg. intended for A & can use the forged keys for authentication.

ii) Publicly available directory

- A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys.
- Maintenance & distribution of the public directory would have to be the responsibility of some trusted entity or organization.

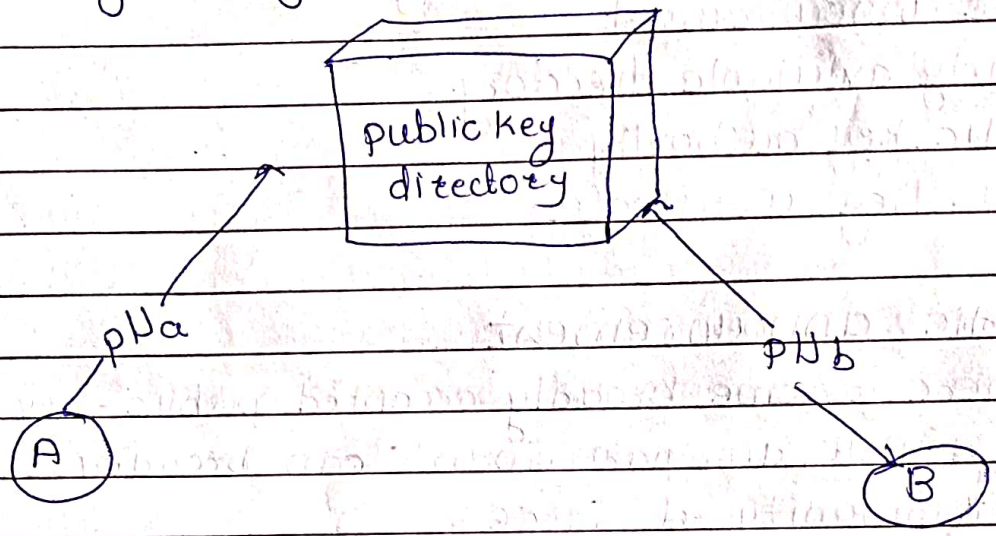


fig- Public key publication

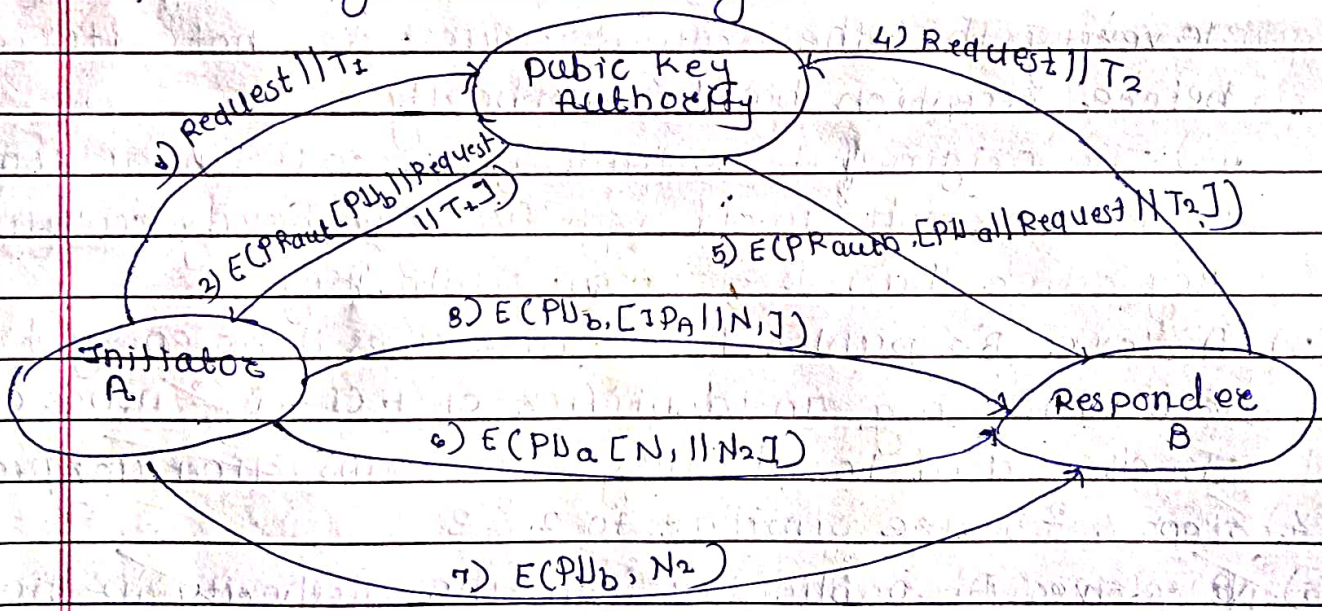
Such scheme would include the following elements.

1) The authority maintains a directory with cname, public key entry of each participant.
2) Each participant registers a public key with the directory authority. Registration would have to be person or by some form of secure authenticated communication.
3) A participant may replace the existing key with a new one at any time, either because of the desire a replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.
4) Participant could also access the directory electronically for this purpose secure, authenticated comm^n from the authority to the participant is mandatory.

- This scheme is clearly more secure than individual public announcement but still has vulnerabilities.
- If anyone succeeds in obtaining the private key of the directory authority then he can pass false public key.
- Another way to achieve the same end is for the adversary to tamper with the records kept by the authority.

iii) Public key authority

- Stronger security for public key distribution can be achieved by providing tighter control over the distribution of public keys from directory.



- Assumes that a central authority maintains a dynamic directory of public keys of all participants.
- Each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.

The foll. steps occur.

1) A sends a timestamped message to the public key authority containing a request for the current public key of B.

2) The authority responds with a message that is encrypted using the authority's private key PRauth. Thus A is able to decrypt the msg. using the authority public key. ∴ A is assured that the msg. originated with the authority. The msg. includes foll.

i) B's public key PU_b, which A can use to encrypt msg destined for B.

ii) The original request used to enable A to match this response with the corresponding earlier request & to verify that the orginal request was not altered before reception by the authority.

iii) The original timestamp given so A can determine that this is not an old msg. from authority containing key other than B's current public key.

3) A stores B's public key & also uses it to encrypt msg. to B containing an identifier of A (ID_A) & nonce (N_1) which is used to identify this transaction uniquely.

4) steps 4 & 5 are similar to 2 & 3

5) B retrives A's public key from the authority in the same manner as A retrived B's public key.

6) B sends msg to A encrypted with PU_a & containing A's nonce (N_1) as well as new nonce generated by B (N_2). Because only B could have decrypted msg. (3), the presence of N_1 in msg (6) assures A that the correspondent is B.

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

iv) Public key certificates :-

- The directory of names & public keys maintained by the authority is vulnerable to tampering.
- In essence, a certificate consists of public key, an identifier of the key owner & whole block signed by trusted 3rd party.
- 3rd party is certificate authority, such as gov. agency or financial institution that is trusted by user community.
- A user can present his/her public key to the authority in a secure manner & obtain certificate.
- The user can then publish the certificate. Anyone needing this use's public key can obtain the certificate & verify i.e valid by way of the attached trusted signature.
- A participant can also convey it's key info. to another by transmitting it's certificate.
- Other participants can verify that the certificate was created by the authority.

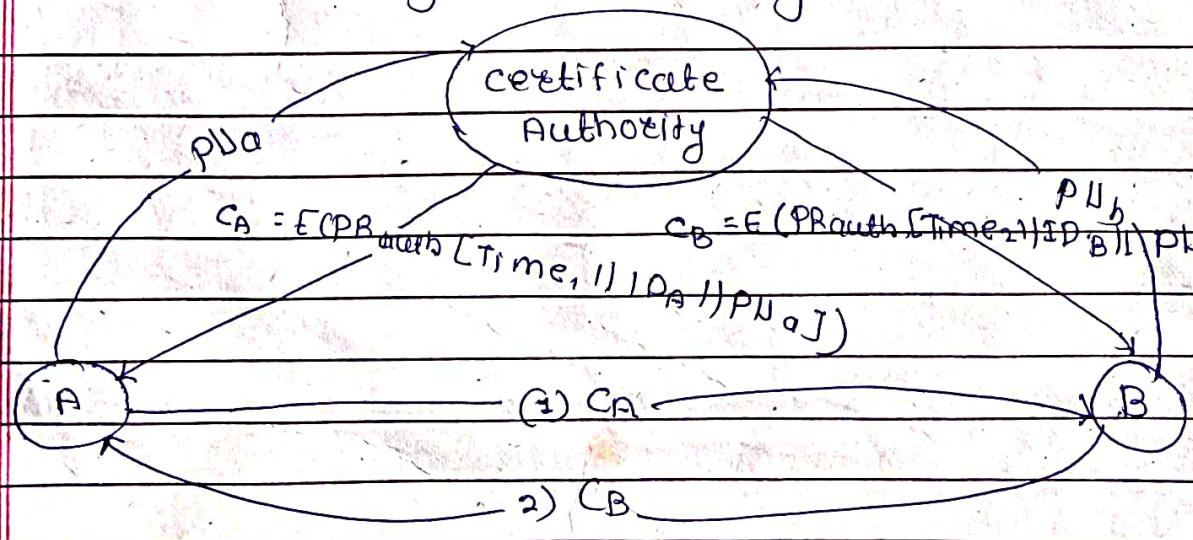


fig. Exchange of public key certificates

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