Mod-3

Distribution of public keys

Several techniques have been proposed for the distribution of public keys

(i) Public announcement
(ii) Publicaly available directory
(iii) public key authority
(iv) public key certificate

(i) Public Announcement

public key encryption is all about using a "public key" that anyone can access.

eg. if you want to send a secure message to someone, you use their public key to encrypt it and only they can decrypt it using their private key

one common method of sharing public keys is to just send them out, like attaching them to emails or posting on online forums.

problem with this approach if someone pretended to be you and shared a fake public key claiming it yours other people would start using that fake key this is called forging a public key

(2) Publicaly available directory

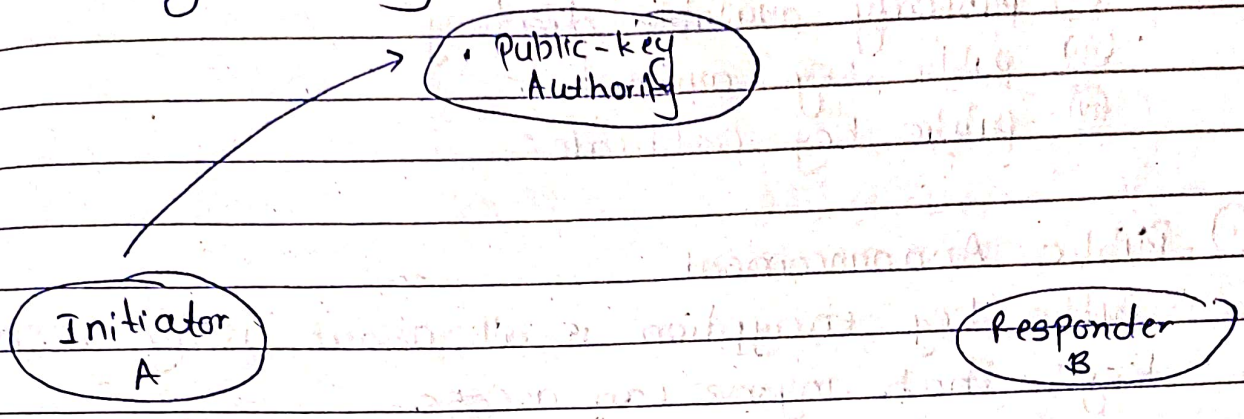
can obtain security by regestering keys with a public directory

directory must be trusted with properties:

- Contain {name, public-key} entries
- participants register securly with directory
- Participants can replace key at any time
- directory is periodically published
- directory can be accessed electronically

still vulnerable to tampering or forgery

③ public key Authority



- improve security by tightening control over distribution of keys from directory
- requires users to know public key for the directory
- users interact with directory to obtain any desired public key securely
- does require real-time access to directory when keys are needed.

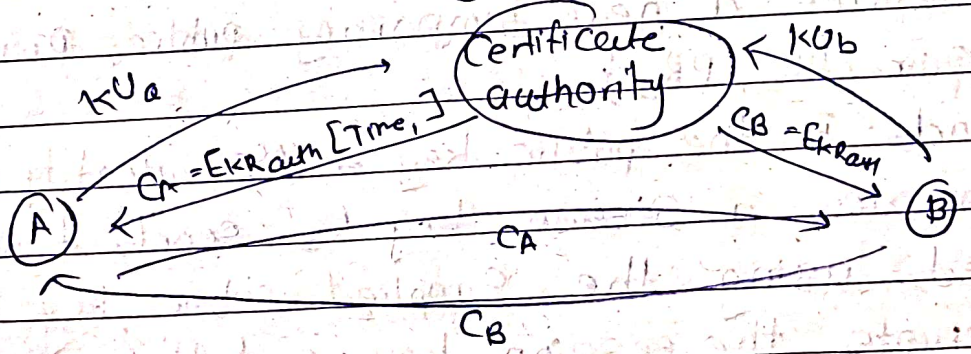
How public key authority works

① A requests B's public key to the authority
② Authority (includes timestamp) sends the response which is encrypted using authority private key, only the authority can encrypt with this key.
A can verify the message authenticity by decrypting it with the authority public key
response contain
B's public key
A's request copy
timestamp

③ A receive message and encrypt message for B.
④ A send message which contain A's ID and a random number (nonce) which is unique to this transaction and ensure the message
⑤ B receive the message it check authetricity by taking A's public key from authority

④ Public key Certificates

- public key authority could be a battleneck in the system
- must appeal to the authority for the key of every other user.
- Certificates allow key exchange without real time access to public key authority
- certificate binds identity to public key with all contents signed by a trusted public-key or certificate authority (CA)



- a user can present his or her public key to the authority in a secure manner and obtain a certificate
- then user can publish certificate

- anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature

Public-key Distribution of secret keys

- public-key algorithms have relatively slow data rates so few users make exclusive use of public key encryption
- usually prefer private-key encryption to protect message contents

- hence session key is needed
  public-key encryption helps distribution of secret keys

① Simple Secret key Distribution

- A generates a new temporary public private key pair $\{PU_a, PR_a\}$
- A sends B the public key & A's identity
- B generates a session key k_s sends it to A encrypted using the supplied public key
- A decrypts the session key & both use

Problem is that an opponent can intercept and impersonate both halves of protocol

Diffie-Hellman key Exchange

- First public-key type schema proposed for key distribution only
- it is a practical method for public exchange of a secret key
- two parties agree on a common key without sharing it directly over the network.
- value of key depends on the participants (and their private and public key info.)

Algorithm

Alice	Bob
Alice & Bob Share a prime no. q and an integer α such that $\alpha < q$ and α is primitive root of q	Alice & Bob share a prime no. q and an integer α such that $\alpha < q$ and α is primitive root of q
Alice generates a private key X_A such that $X_A < q$	bob generates a private key X_B such that $X_B < q$
Alice Calculates a public key $Y_A = \alpha^{X_A} \bmod q$	Bob Calculates a public key $Y_B = \alpha^{X_B} \bmod q$
Alice receives Bob's public key Y_B in plain-text	Bob receives Alice's public key Y_A in plaintext
Alice Calculates Shared secreat key $K = (Y_B)^{X_A} \bmod q$	Bob Calculates shared secreat key $K = (Y_A)^{X_B} \bmod q$

K is used as session key in private key encryption schema

if alice & Bob subsequently communicate, they will have the same key as before, unless they choose new public-keys

eg

- Alice & Bob agree on prime no. $q = 353$, $\alpha = 3$
- select random private key

$x_A = 97$ $\quad$ $x_B = 233$

$Y_A = (3)^{97} \bmod 353 = 40$ $\quad$ $\alpha^{x_A} \bmod q$

$Y_B = (3)^{233} \bmod 353 = 248$ $\quad$ $\alpha^{x_B} \bmod q$

compute shared session key

$K_{AB} = (Y_B)^{x_A} \bmod q$

$= (248)^{97} \bmod 353$

$= 160$

$K_{AB} = (40)^{233} \bmod 353$

$= 160$