Key Distribution

- Symmetric Key Distribution
  - Symmetric Encryption
  - As-symmetric Encryption
    - Simple Secret Key Distribution
    - Secret Key Distribution with Confidentiality & Authentication
- Public Key Distribution
  - Public Announcement of public keys
  - Publicaly Available Directory
  - Public key authority
  - Public key certificates

Symmetric Key Distribution

Symmetric Encryption

*) Process between two parties that exchanges the key

*) Two parties / person-A and person-B can exchange the keys in the following ways.

Physical Meet

Advantage : More secured

Disadvantage : More time

A can physically Deliver to B

Key Distribution Center [KDC]

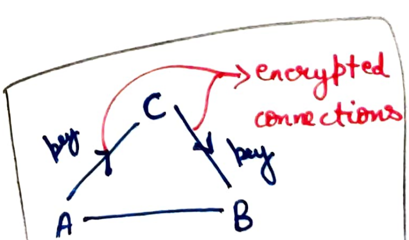
Generate the keys and send to the users involved.

Advantage : Authentic, but should rely on 3rd party.

Using Previous keys :

Old keys ← encryption and generate new keys

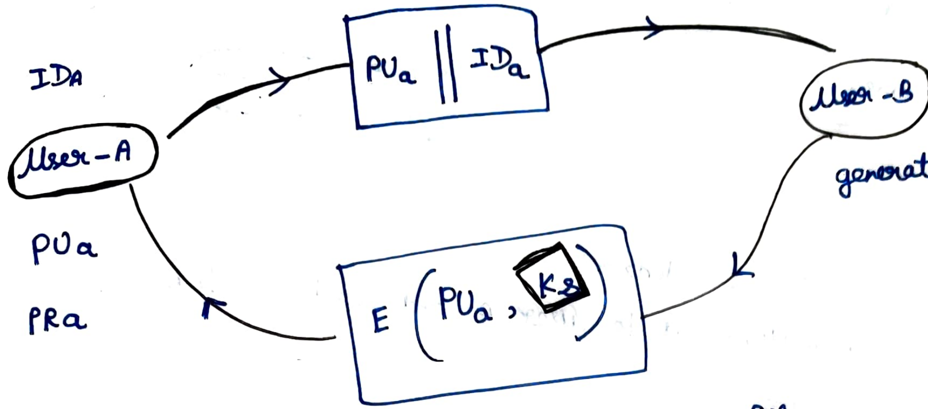
Using third Party :



No Confidentiality
No Authentication

As-symmetric Encryption

Simple Secret key Distribution



*) Now user-A decrypts using PRa and discards PUa

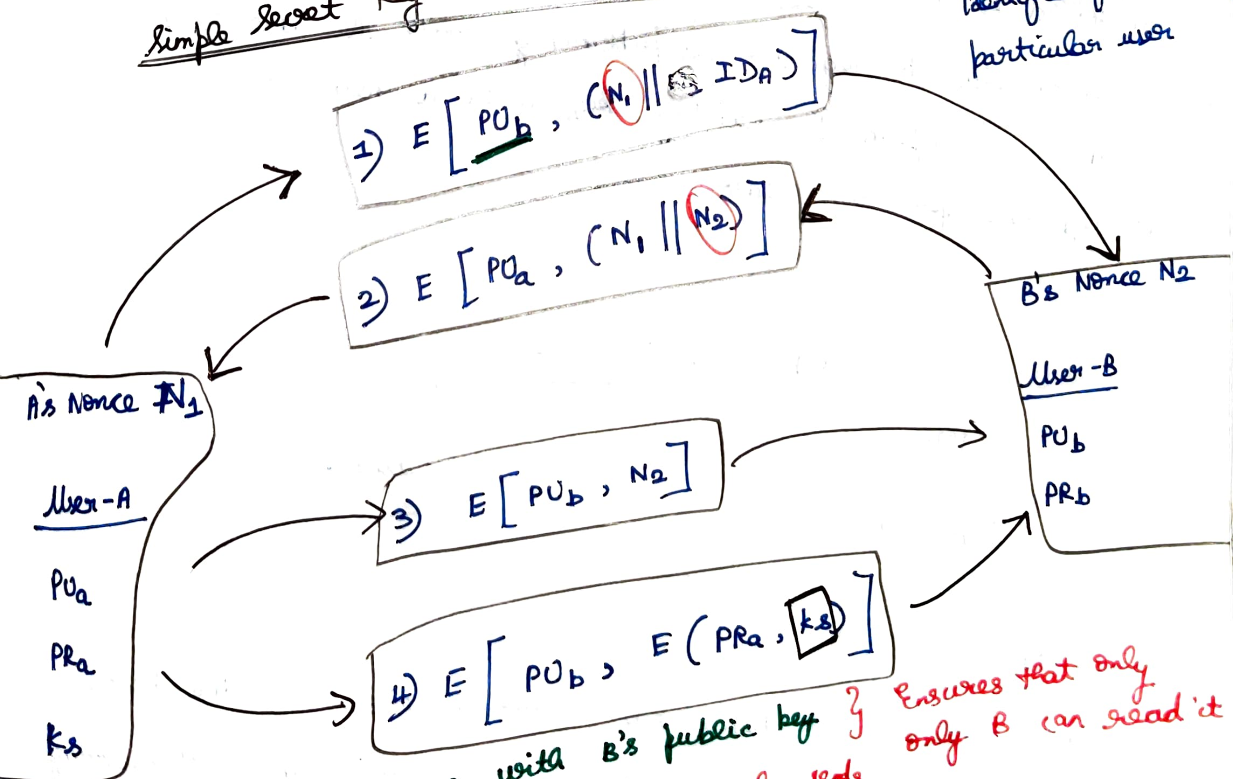
*) Now user-A has got the secret key Ks

*) User-A and User-B has the secret key Ks. From now, communication starts

Simple Secret Key Distribution

confidentiality
authentication

Nonce ⇒ unit identifier of that particular user

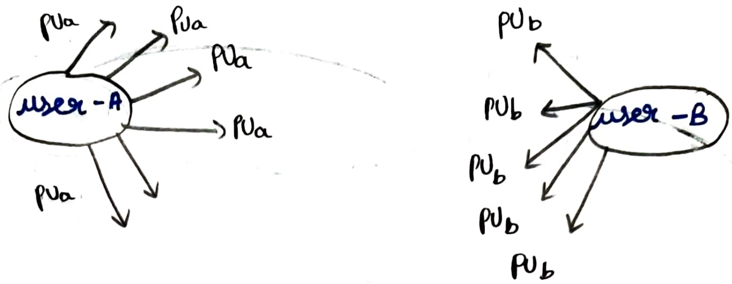


*) Encryption of the message with B's public key } Ensures that only only B can read it

*) Encryption with A's private key } A only sends to B

Public key Distribution

1) Public Announcement of Public keys



*) The Public keys are broadcasted.

*) Any user can pretend to be user-A and send the public key to another user.

*) Until user-A has got this thing and alerts to the other user, a pretender is able to read all encrypted messages of other users.

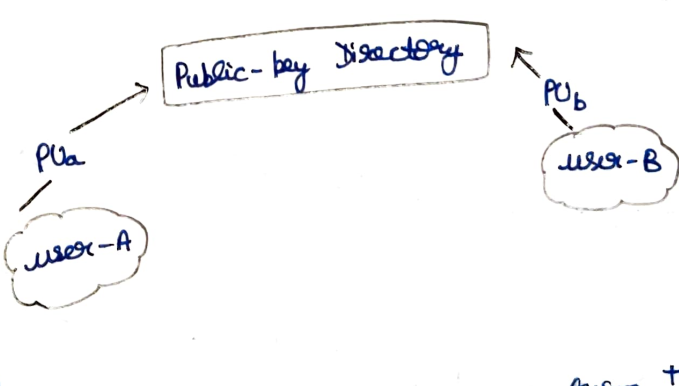
2) Publicly Available Directory

*) A dynamicly available directory which is used to achieve the security.

*) Maintenance and Distribution of these public keys is controlled by a trusted entity.

*) Each user has to register the public key with the directory.

*) A user can replace the existing key with a new one at any time for any reason

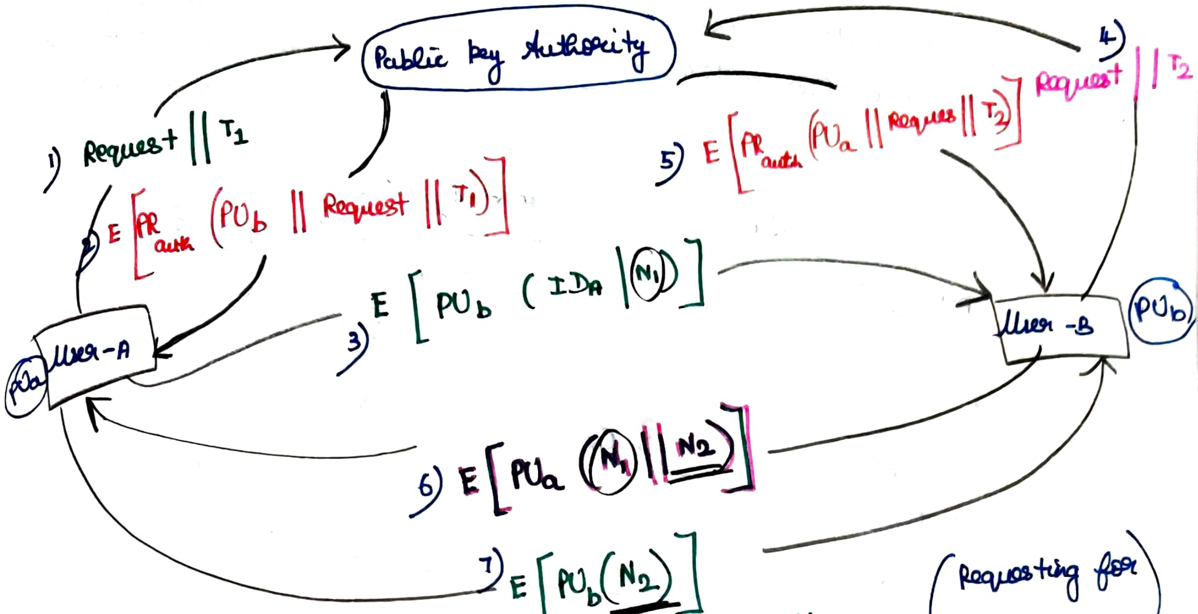


Disadvantage

*) User B fetched the public key of user-A from the directory. and immediately user-A updates the public key ⇒ Mis-leading

*) Any un-authorized person can store their public key.

3) Public Key Authority



(PKA) (Requesting for PU_b)

1) user-A to Public Key Authority ⇒ Request || T_1

2) PKA to user-A ⇒ $E[PR_{auth} ; (PU_b \| Request \| T_1)]$

Now user-A with its PU_{auth} (decrypts) and uses PU_b to send message to user-B.

N ⇒ Nonce which gives unique transaction | authentication

3) user-A to user-B $E[PU_b ; (ID_A \| N_1)]$

Now user-B with its PU_{auth} decrypts ~~and~~ note down its ID_A and N_1

4) Now user-B wants to send message to user-A.

user-B to PKA ⇒ Request || T_2

5) PKA to user-A ⇒ $E[PR_{auth} ; (PU_a \| Request \| T_2)]$

Now user with its PU_{auth} (decrypts) and uses PU_a to send message to user-A

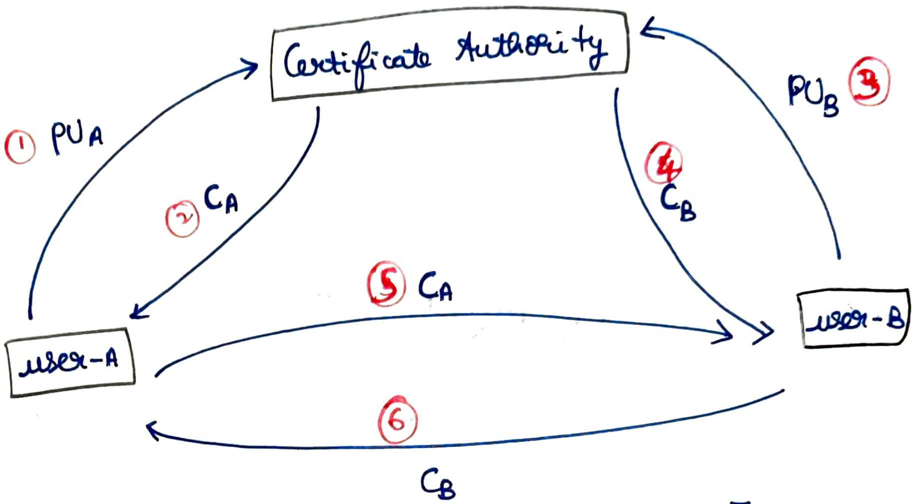
6) user-B to user-A ⇒ $E[PU_a , [N_1 \| N_2]]$ [Reply to step-3]

7) user-A to user-B ⇒ $E[PU_b (N_2)]$ (acknowledgement purpose)

4) Public Key Certificate Authority

[Before the communication starts they will exchange their certificates]

Includes public key and their identity



Certificate of user-A $\Rightarrow E\left[PR_{auth}\left(PU_A \| ID_A \| T_1\right)\right]$

Certificate of user-B

$\Rightarrow E\left[PR_{auth}\left(PU_B \| ID_B \| T_2\right)\right]$

Public key Authority
&
Public key Certificate Authority

} Trusted 3rd party