Chapter Seven

Key Management

Key Management

Public-key cryptosystem helps to solve key distribution problems

Two aspects of key management:

- distribution of **public keys**
- use of public-key cryptosystem to distribute **secret keys**

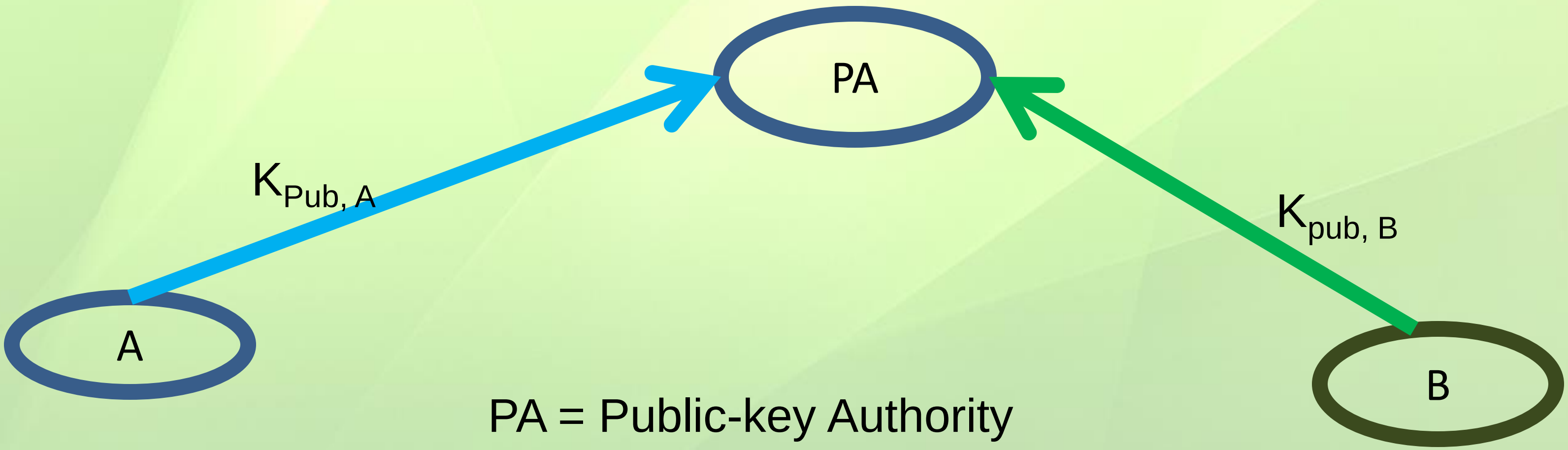
Distribution of Public Keys

- Distribution of public keys can be performed using:
  - public-key authority
  - public-key certificates

Public-Key Distribution by PA

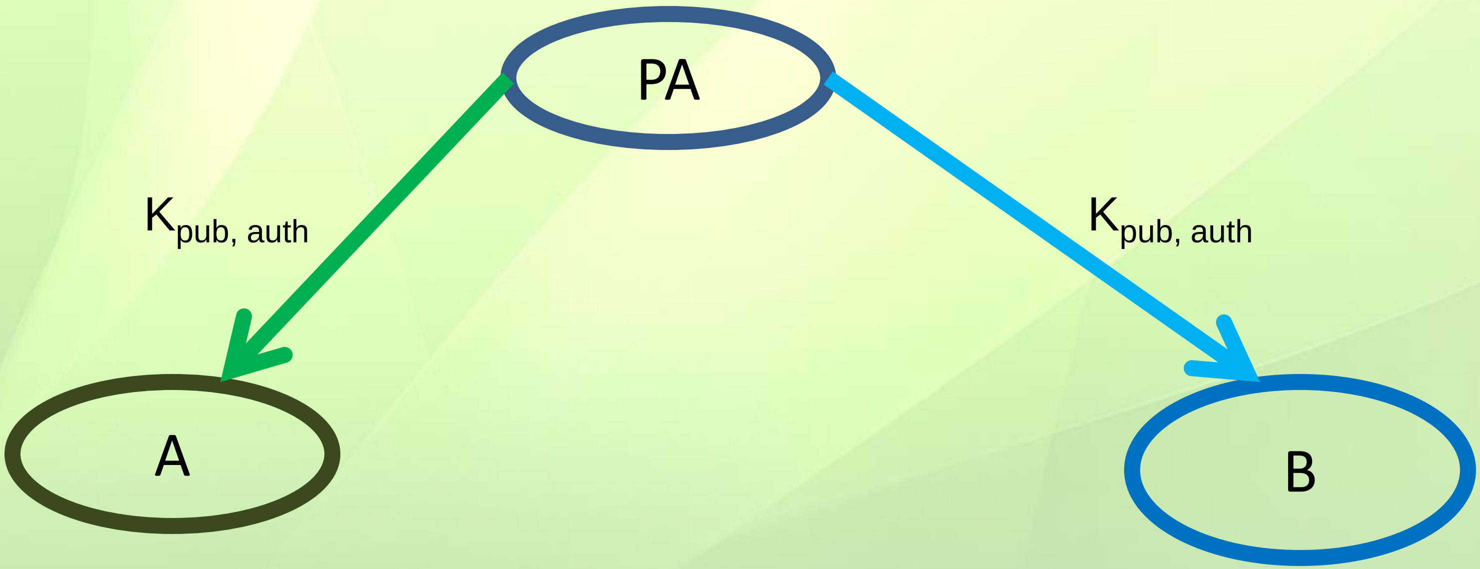
If A and B want to communicate each other, they will register their public keys ($K_{pub, A}$ and $K_{pub, B}$) to the public authority.

Thus the authority can send encrypted message to A as well as to B.



Public key distribution

- After registration PA gives them (A and B) the public key of the authority.



Public Key Distribution by PA

- Suppose **A** needs current public key of **B**.

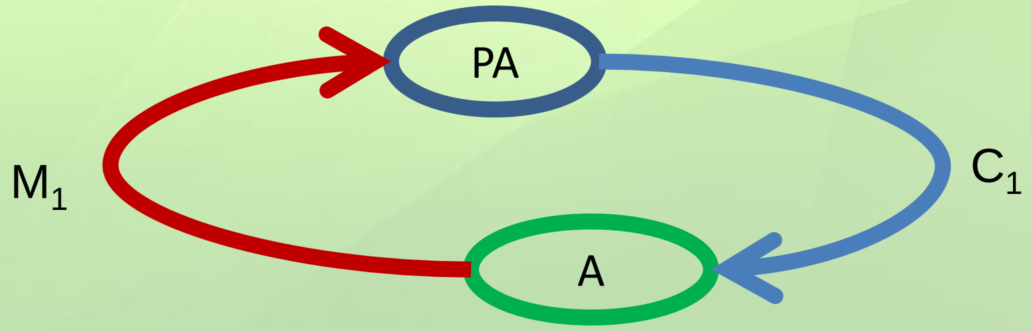
1) A sends a message to PA (public-key authority) as follows:

$$M_1 = \text{Request} \,||\, \text{Time1}.$$

2) The authority sends cipher text to A:

$$C_1 = E\,(K_{p,\ auth},\ [\ \mathbf{K_{pub,\ B}}\ ||M_1])$$

Where $K_{p,\ auth}$ is the private key of the authority.



Public key Distribution By PA

- A will decrypt C_1 using the public key of the authority and get the public key of B:

$$P_1 = D\ (K_{pub,Auth},\ C_1)$$

$$P_1 = K_{pub,\ B}\ ||M_1$$

A got the public key of the authority after the registration of his public key to the authority.

Public Key Distribution

3) Now A will send an encrypted message to B as follows:

$$C_2 = E\ (K_{pub,B},\ [ID_A \,||\, \mathbf{N_1}])$$

Where N_1 is called nonce, which is generally a random number.

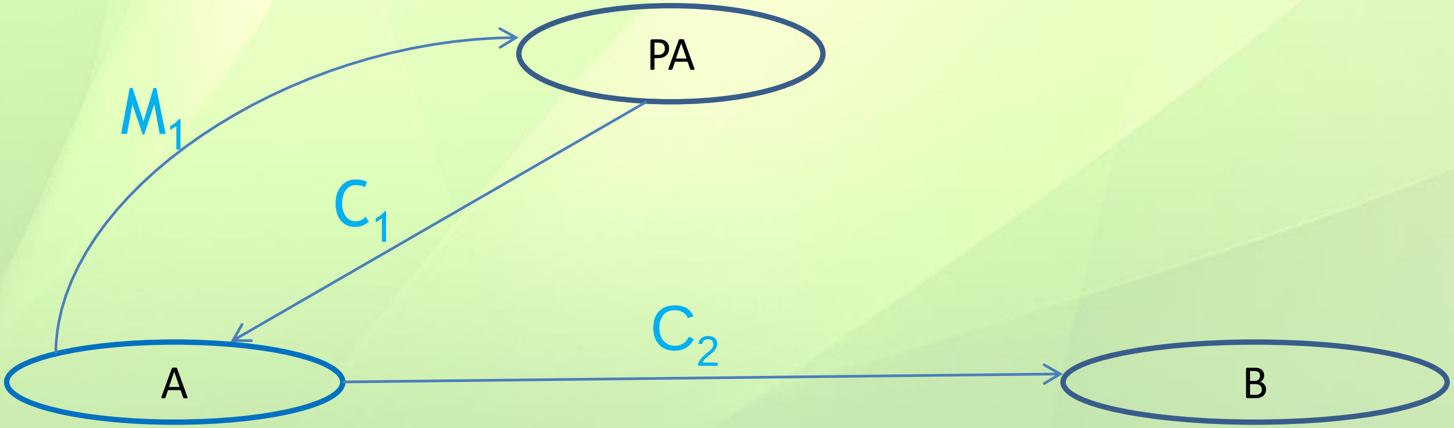
A will send C_2 to get a response from B to ensure that $K_{pub,B}$ is the public key of B. Now B should say 'yes' it belongs to him.

Previous Three Steps at a Glance

1) M_1 = Request || Time1. [Request from A]

2) $C_1 = E\,(K_{p,\,auth},\ [\,\mathbf{K_{pub,\,B}}\,||M_1])$ [Response from PA]

3) $C_2 = E\,(K_{pub,B},[ID_A\,||\,\mathbf{N_1}])$ [Cipher Message from A]



Public Key Distribution

To give response to **A**'s message, **B** should send an encrypted message. So that masqarand can not know his public key. To send an encrypted message to **A**, **B** has to know the public key of **A**.

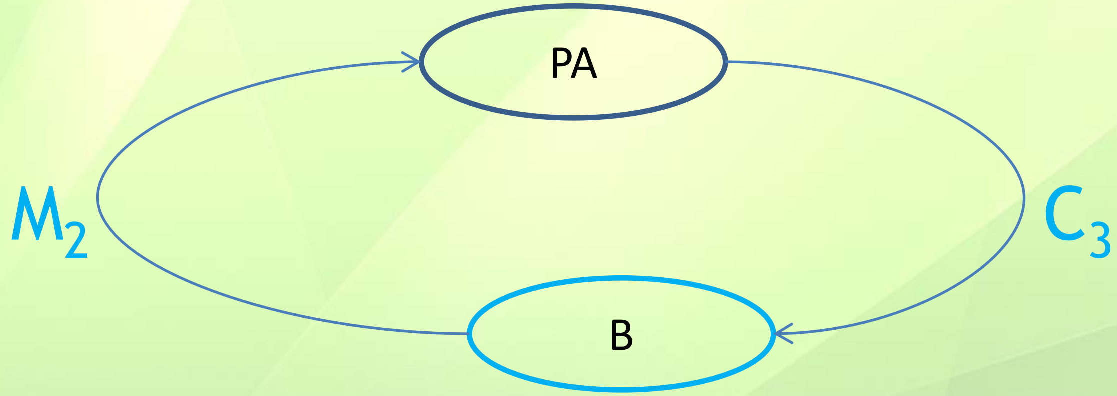
4) Now **B** sends a request to the authority as follows:

M_2 = Request || Time 2

Public key Distribution

5)The authority sends an encrypted message to **B** as follows:

$C_3 = E\,(K_{p,\ auth},\ [\mathbf{K_{pub,\ A}}\ ||M_2])$



By decrypting the message **B** gets the public key of **A**.

Public Key Distribution

6) **B** responds to **A** with an encrypted message as follows:

$C_4 = E\ (K_{pub,\ A},\ [N_1 || \mathbf{N_2}])$

A decrypts C_4 and get N_1, which proves that **B** sends C_4 as response. Because no one except **B** knows N_1,whis was sent by **A**.

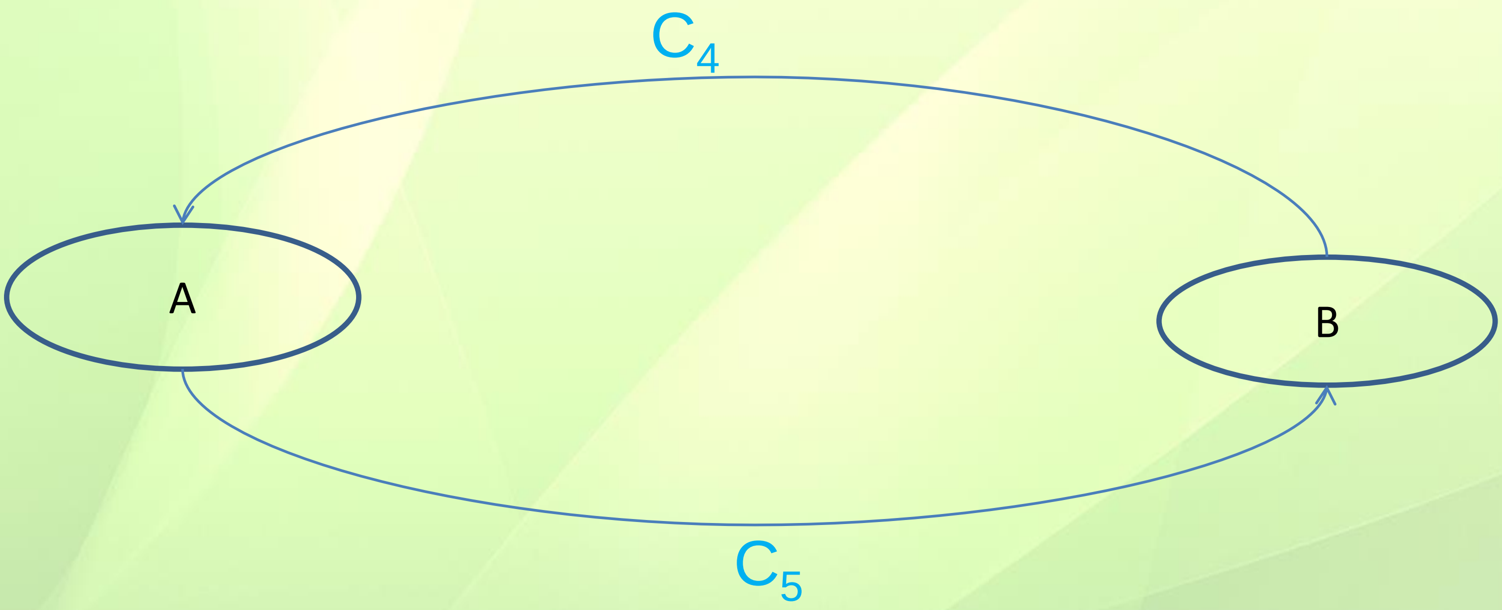
7) **A** responds to **B** as follows:

$C_5 = E\ (K_{pub,\ B},\ \mathbf{N_2})$

Similarly **B** decrypts C_5 and became sure that the message was from **A** (by getting N_2).

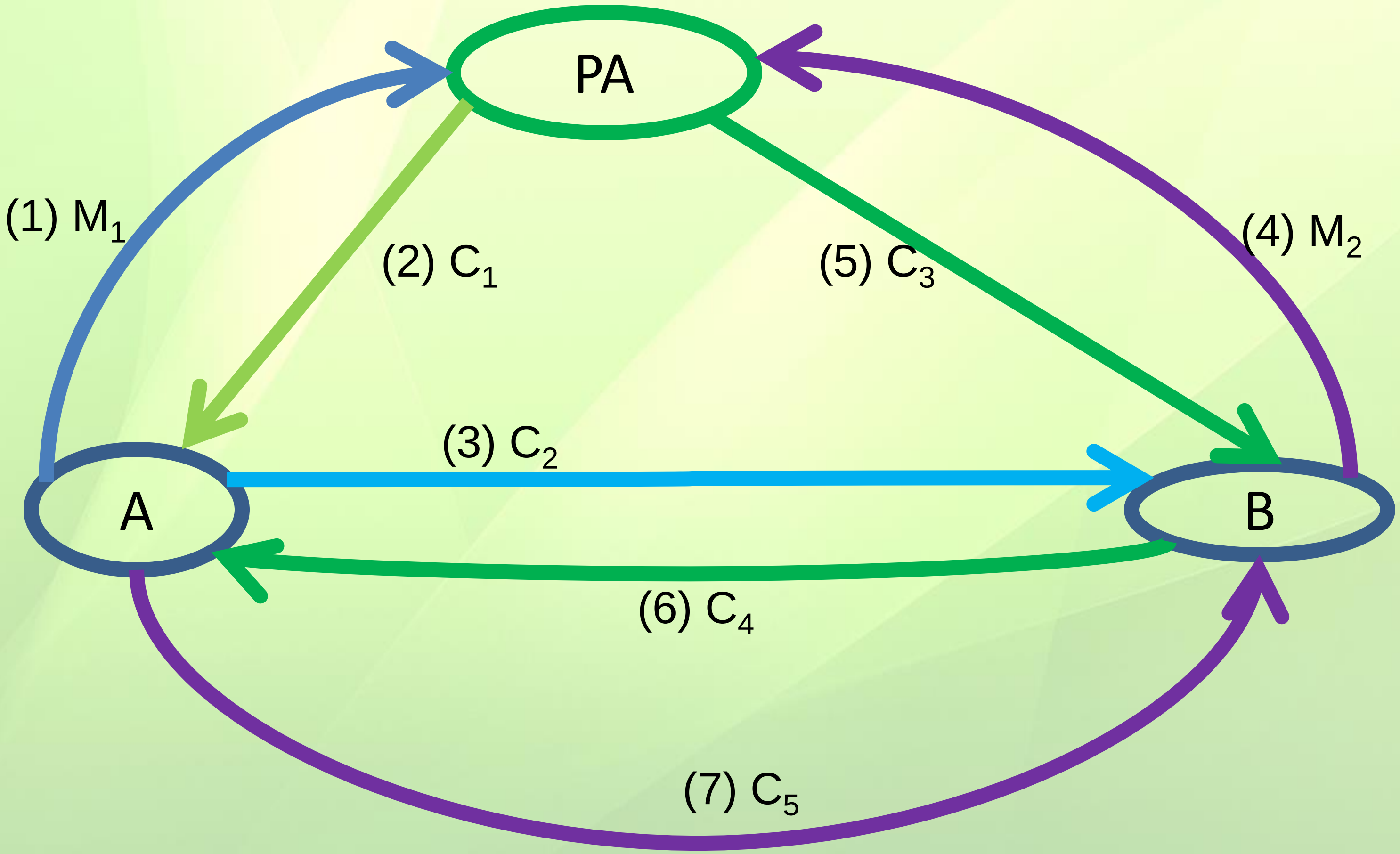
Public Key Distribution

$C_4 = E\ (K_{pub,\ A},\ [N_1 || \mathbf{N_2}])$



$C_5 = E\ (K_{pub,\ B},\ \mathbf{N_2})$

Public Key Distribution at a Glance



Public Key Distribution

(1) M_1 = Request || Time1. $[A \xrightarrow{\text{Request}} PA]$

(2) $C_1 = E(K_{p,\ auth}, [\ \mathbf{K_{pub,\ B}} \,||\, M_1])$ $[PA \xrightarrow{\text{Response}} A]$

(3) $C_2 = E(K_{pub,B}, [ID_A \,||\, \mathbf{N_1}])$ $[A \xrightarrow{\text{Message}} B]$

(4) M_2 = Request || Time 2 $[B \xrightarrow{\text{Request}} PA]$

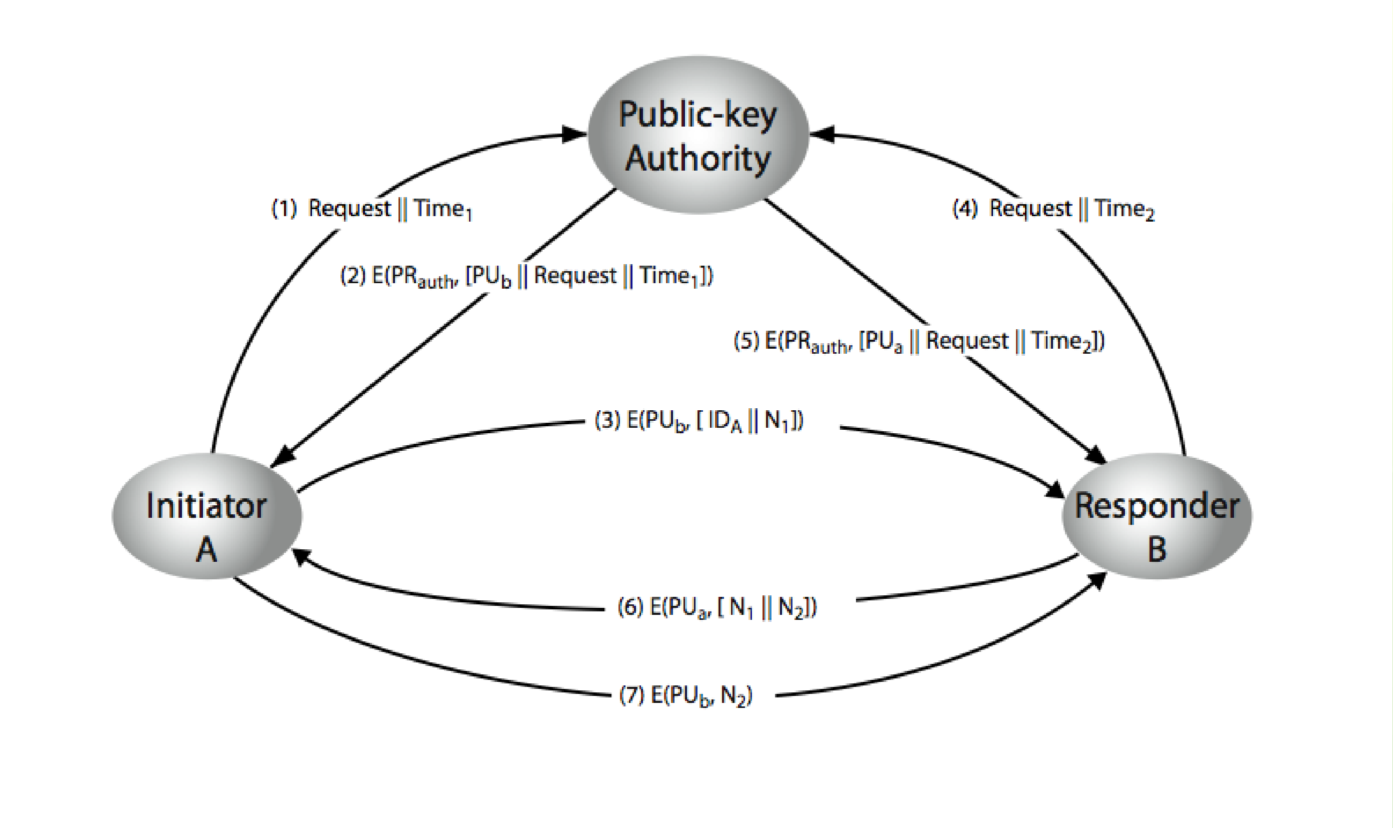
(5) $C_3 = E(K_{p,\ auth}, [\mathbf{K_{pub,\ A}} \,||\, M_2])$ $[PA \xrightarrow{\text{Response}} B]$

(6) $C_4 = E(K_{pub,\ A}, [\mathbf{N_1} \,||\, \mathbf{N_2}])$ $[B \xrightarrow{\text{Reply}} A]$

(7) $C_5 = E(K_{pub,\ B}, \mathbf{N_2})$ $[A \xrightarrow{\text{Reply}} B]$

Public-Key Authority

PR = Private key, PU = Public key



Public-key Distribution through Certificate Authority

User A registers his public key to the certificate authority (CA).

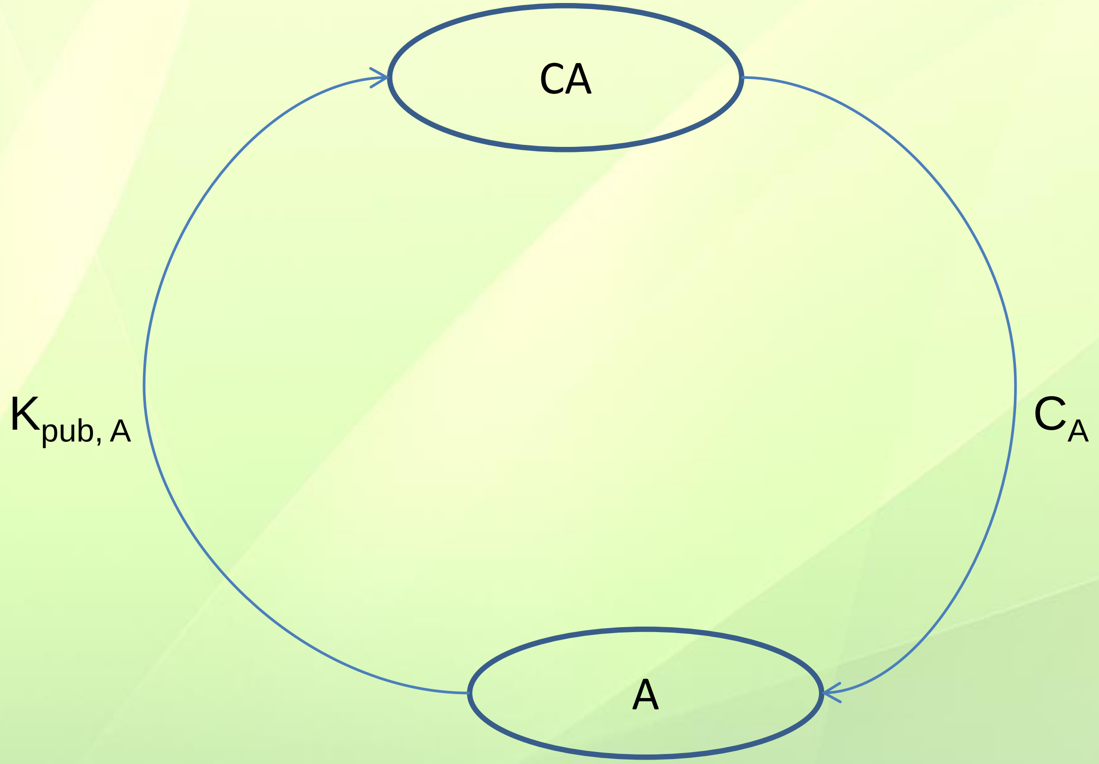
The CA sends a certificated (encrypted message containing public key of A) to A.

Similarly B registers his public key to the CA.

The CA sends a certificate to B.

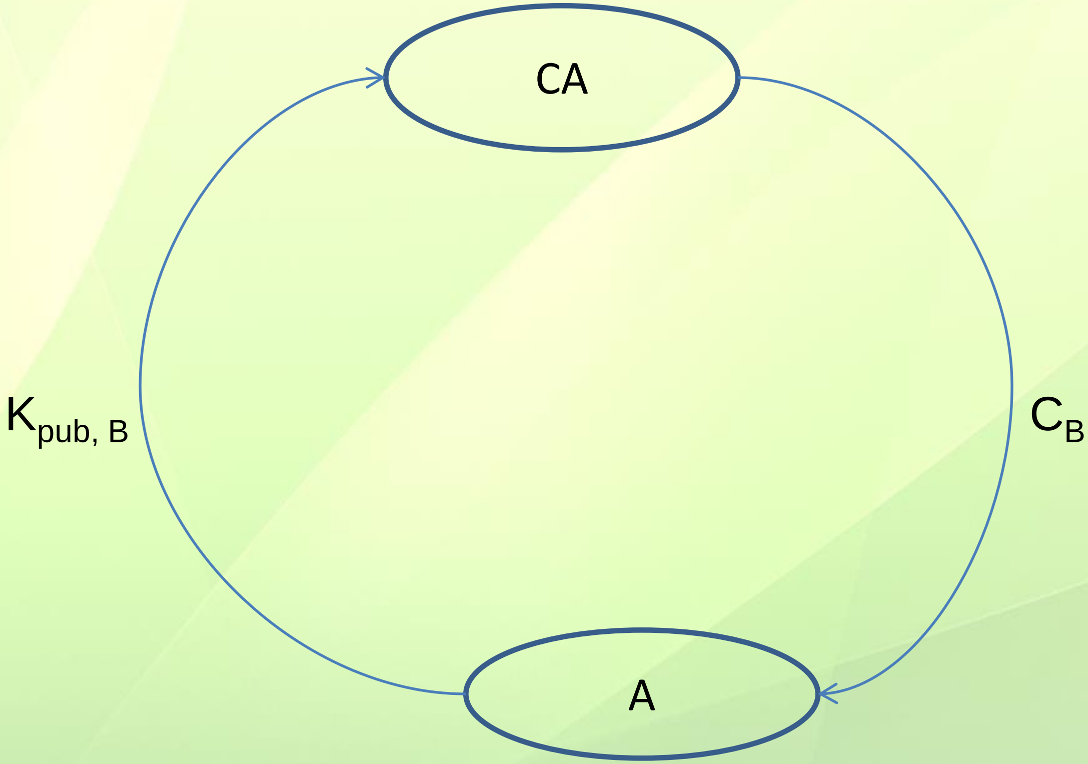
If A and B wants to share secret message, A and B will exchange their certificate.

Public-key Certificate



$C_A = E(K_{p, Auth}, [T_1 \| ID_A \| \mathbf{K_{pub, A}}])$

Public Certificate

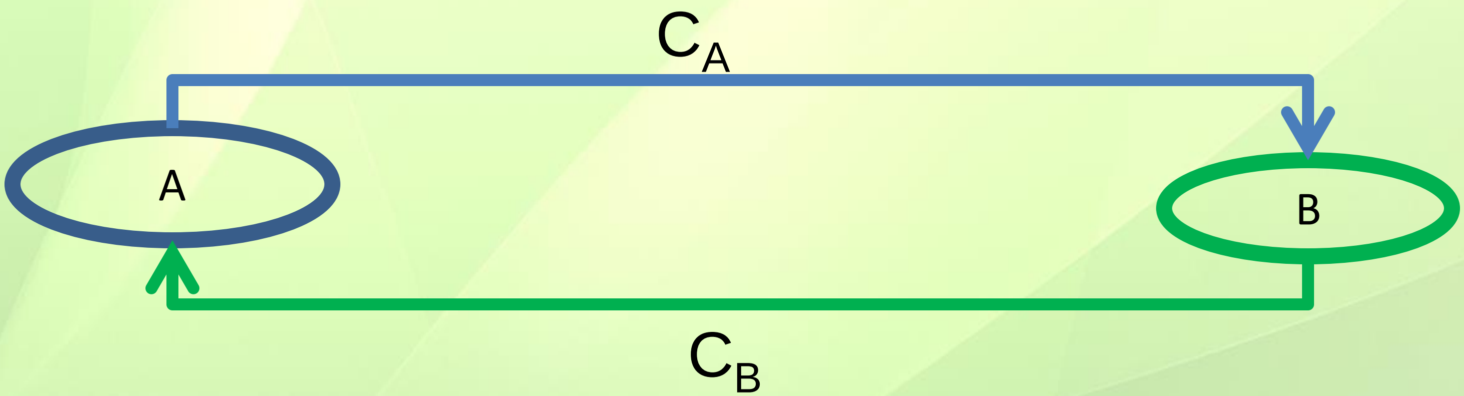


$C_B = E(K_{p, Auth}, [T_1 \,||\, ID_B \,||\, \mathbf{K_{pub, B}}\,])$

Public key certificate

$C_A = E(K_{p,\ Auth}, [T_1 \,||\, ID_A \,||\, \mathbf{K_{pub,\ A}}])$

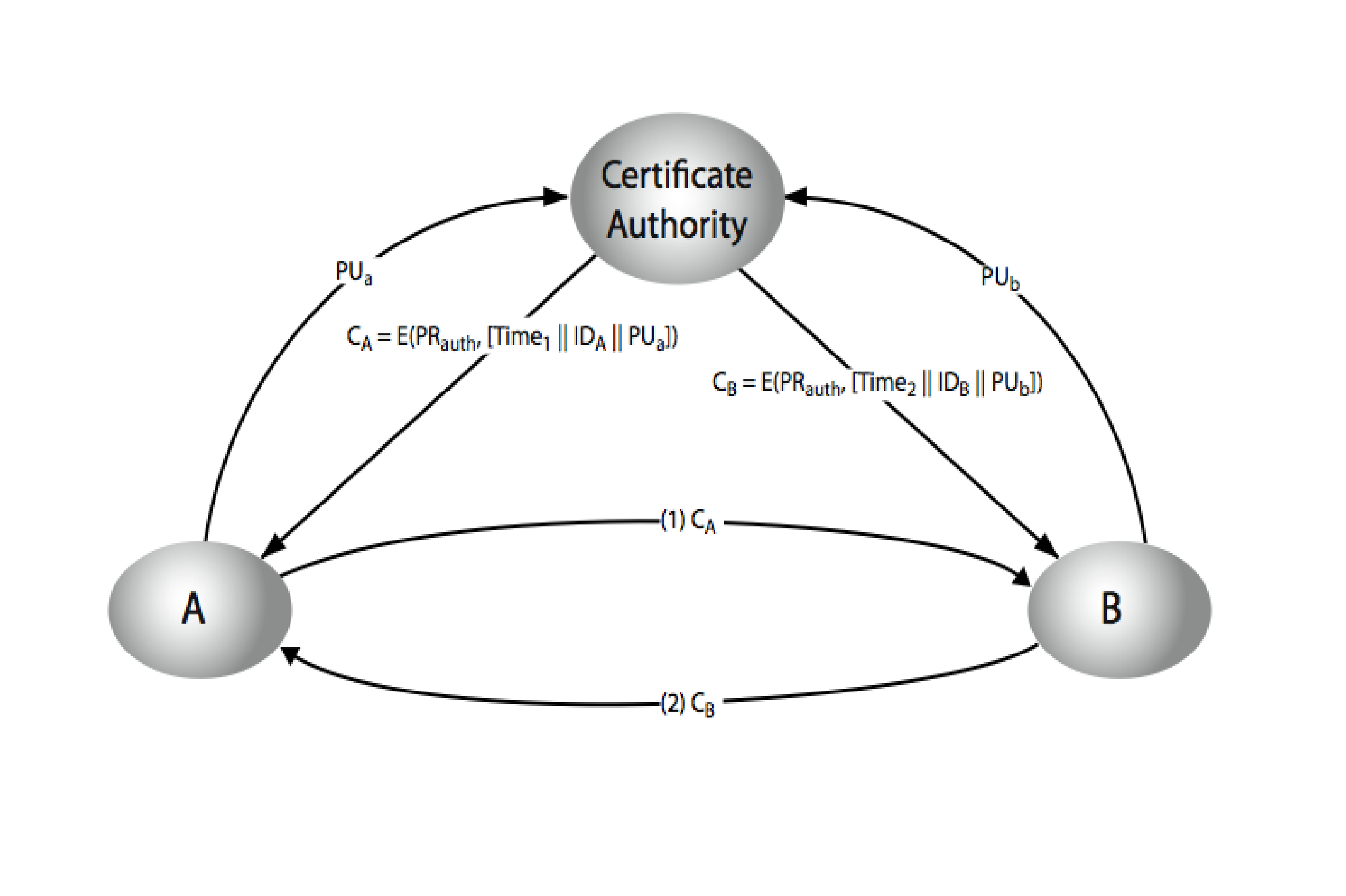
$C_B = E(K_{p,\ Auth}, [T_1 \,||\, ID_B \,||\, \mathbf{K_{pub,\ B}}])$



Both A and B knew the public key of CA and they can decrypt the certificates (C_A, C_B). Thus they will get the public key of each other.

Public-Key Certificates

PU = Public key, PR = Private key



Secret key distribution

- In symmetric or private key cryptosystem both parties **share the same key** for encryption and decryption.
- Secret or private key can be distributed using public key encryption. The following is the way of distribution:

Suppose A and B wish to share the secret. If key and the key is in possession of A, then A encrypts the secret key using public key of B and sends to B. B decrypts the message and gets the secret key.

Secret key distribution

- However in this process there is a drawback: **how B ensures that the encrypted message is from A.**

So, to share the secret key at first A and B will authenticate each other. Next A will send secret key in cipher text (encrypted secret key) to B.

Secret key distribution

- The process is as follows:

1) A sends an encrypted message to B.

$C_1 = E(K_{pub,\,B}, [\mathbf{N_1} \,||\, ID_A]).$

2) B responds as sending another cipher text

$C_2 = E(K_{pub,\,A}, [\mathbf{N_1} \,||\, \mathbf{N_2}])$

Where N_1 and N_2 are nonce.

Secret key distribution.

- A responds with the following cipher text

  3) $C_3 = E(K_{pub,\ B}, \mathbf{N_2})$.

A now sends the encrypted secret key in cipher text as follows:

4) $C_4 = E(K_{pub,.B},[\ \mathbf{N_1}\ ||\ K_s])$.

Where K_s is the secret key.

Secret key distribution at a glance

1) $C_1 = E(K_{pub, B}, [N_1 || ID_A])$ [Initiative from A]

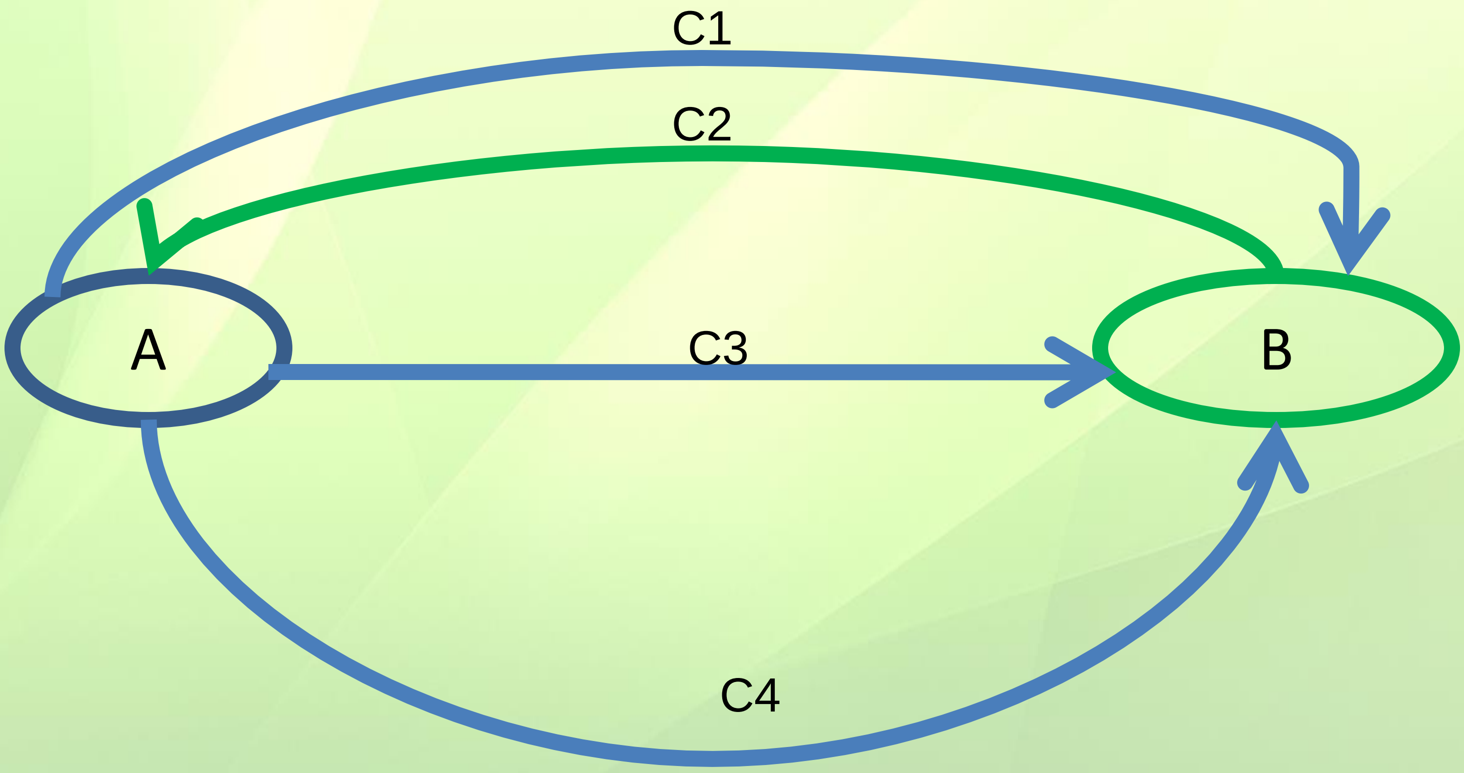
2) $C_2 = E(K_{pub, A}, [N_1 || N_2])$ [Response from B]

3) $C_3 = E(K_{pub, B}, N_2)$ [Response from A]

4) $C_4 = E(K_{pub,.B}, [N_1 || K_s])$ [Key from A].

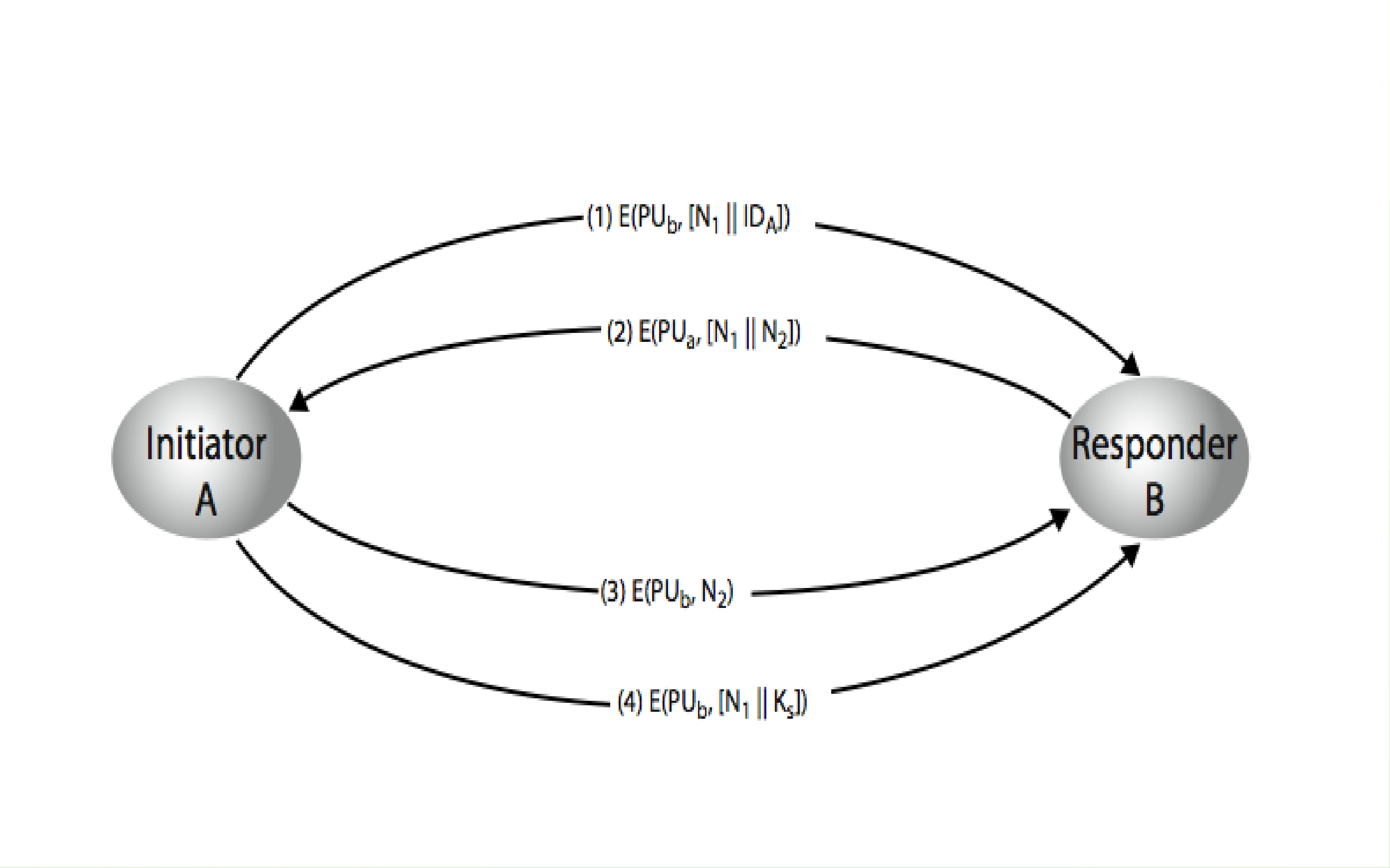
B recognizes A by ID_A. A recognizes and authenticates B by N_1. B authenticates A by N_2. After recognition and authentication A sends secret key.

Pictorial view of key distribution



Distribution of Secret Key

PU = Public key, K_s = Secret key



28

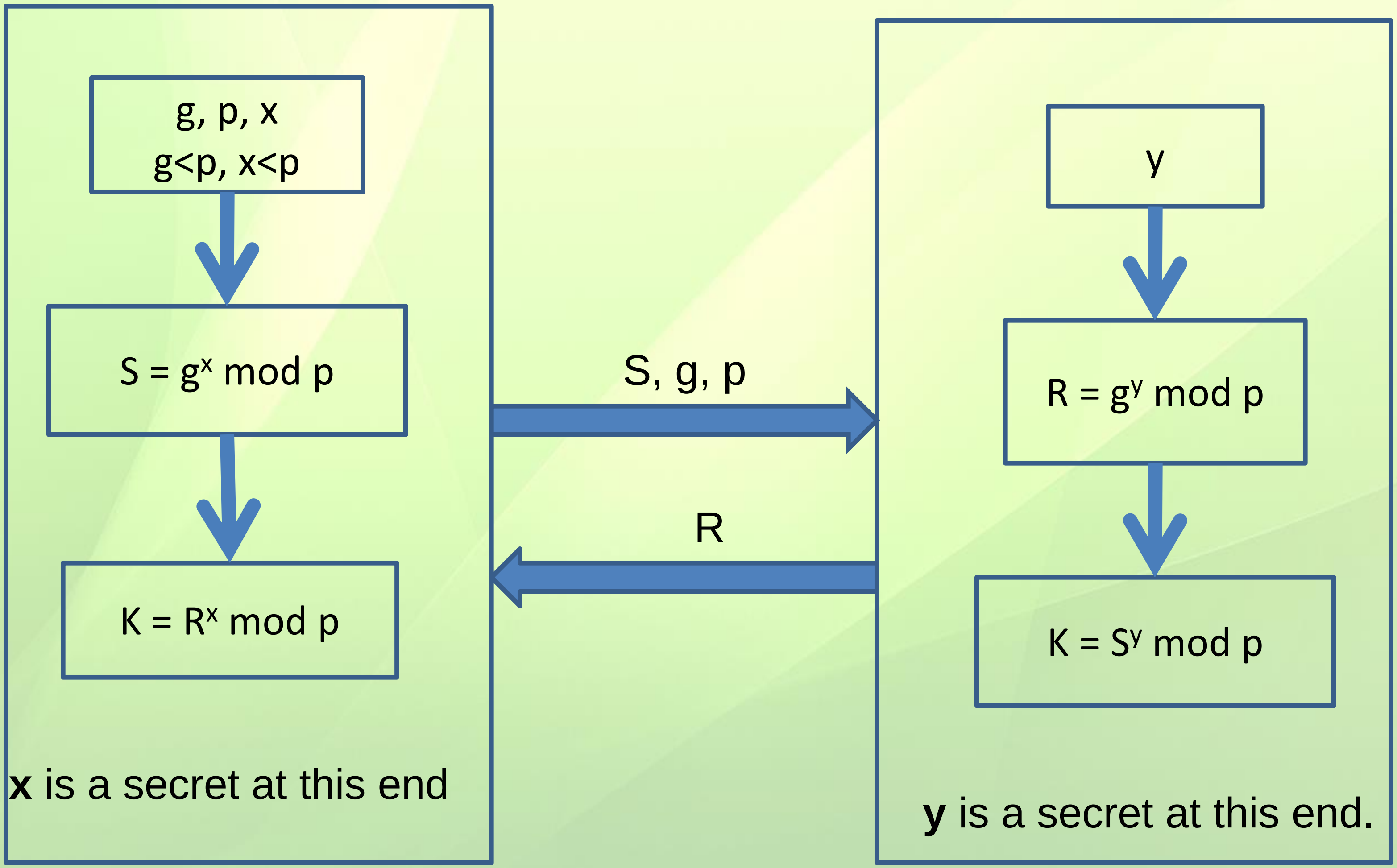
Diffie-Hellman (D-H) key exchange

The scheme was first publicly published by W. Diffie and M. Hellman in 1976.

It is a cryptographic protocol that allows two parties without prior knowledge to share secret key over an insecure communication channel.

The synonym of Diffe-Hellman key exchange scheme is Diffie-Hellman key agreement scheme.

Diffie-Hellman Key Exchange



Diffie-Hellman Key exchange

- **Verification:**

1) $K = S^y \bmod p = (g^x \bmod p)^y \bmod p = \mathbf{g^{xy} \bmod p}$

2) $K = R^x \bmod p = (g^y \bmod p)^x \bmod p = \mathbf{g^{xy} \bmod p}$

Diffie-Hellman key Exchange

❑ Example:

Suppose A and B share the secret.

1) Select p = 29 and g = 2

2) A chooses a **secret** integer x = 8 and computes

$$S = g^x \bmod p = 2^8 \bmod 29 = 24$$

3) B choose a **secret** integer y = 18 and computes,

$$R = g^y \bmod p = 2^{18} \bmod 29 = 13$$

4) A computes, $K = R^x \bmod p = R^8 \bmod 29$

$$= 13^8 \bmod 29 = \mathbf{16}$$

5) B computes, $K = S^y \bmod p = 24^{18} \bmod 29 =$ **16.**

Thank You.