BCT 2306:Fundamentals of Computer Security and Technology

Course Outline

- Information security in computer and communication systems
- Threat analysis (under security attack)
- Security services
- Physical and logical security
- Access control
- Identification
- Authentication
- Authorization
- Programmed threats
- Internet connection policies and implementation
- Message confidentiality
- Message authentication
- Non repudiation
- Web security
- Classical and public key cryptosystems
- Application to information schemes and digital signatures
- Key distribution and key agreement
- Authentication and secret sharing
- Security policies

Lecture 6
Key Distribution and Key Agreement

Introduction

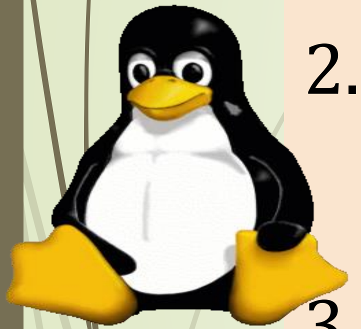
Definitions

Key distribution: A mechanism whereby one party chooses a secret key and then transmits it to another party or parties.

Key agreement: Denotes a protocol whereby two (or more) parties jointly establish a secret key by communicating over a public channel.

In a key agreement scheme, the value of the key is determined as a function of inputs provided by both parties.

Key Distribution



1. Symmetric encryption schemes require both parties to share a common secret key

   Issue is how to securely distribute this key without revealing it to an adversary

2. Many attacks are based on poor key management and distribution

   Rather than breaking the codes

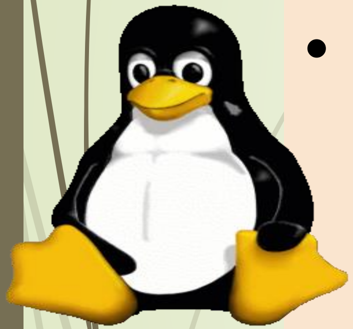
3. This is, actually, the most difficult problem in developing secure systems

Key Distribution

Various key distribution alternatives exist for parties A and B:

1. A can select key and physically deliver to B
   - Does not scale for a large and distributed group
   - How many keys do we need for N users?
2. Third party can select & physically deliver key to A & B
   - Similar comment as 1
   - Sometimes finding a “trusted” third party is another problem
3. If A & B have communicated previously, they can use previous key to encrypt a new key
   - Good option but initially several keys to be distributed
4. If A & B have secure communications with a third party C, C can relay key between A & B on demand
   - Only N master keys are enough

Key Distribution Facts



- *"Conservation of trust"* principle

  - A secure communication cannot be based on nothing; either there should be an initial direct contact or an indirect protocol to transfer trust
- Either physical delivery or a trusted third party

  - Physical delivery is the only option to avoid a third party

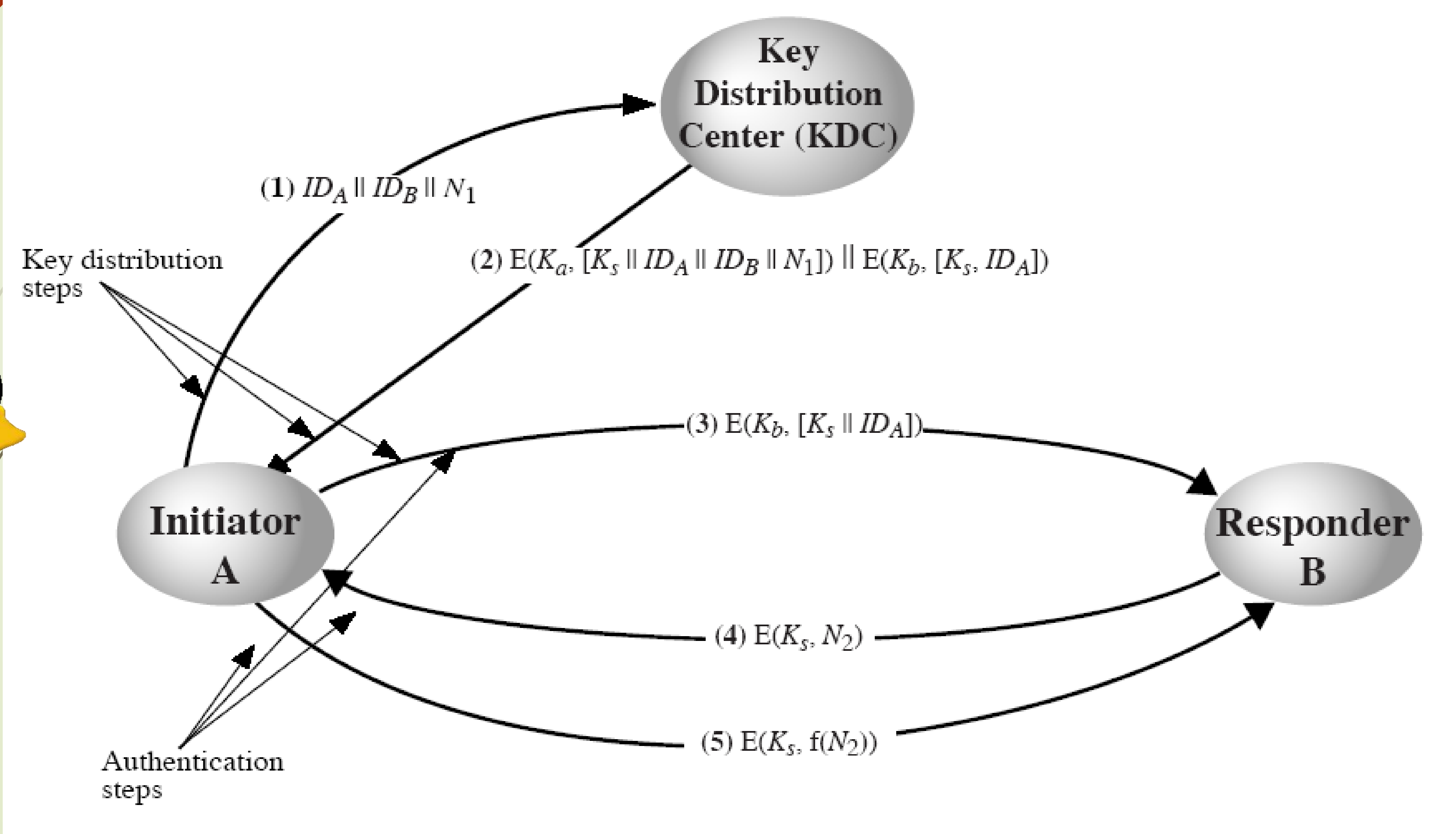
    - Most basic system is PIN entry

  - Otherwise regardless of symmetric or asymmetric encryption, you need a trusted third party

Key Distribution Centre (KDC)

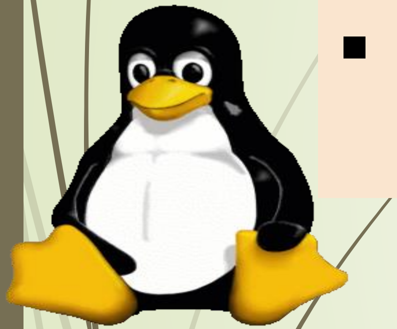
- The use of a KDC is based on the use of a hierarchy of keys. At minimum, two levels of keys are used.
- Communication between two end systems is encrypted using a temporary key often referred to as a session key.
- Typically, the session key is used for the duration of a logical connection and then discarded
- The master key is shared between the KDC and an end system or user and is used to encrypt the session key.

Key Distribution Scenario

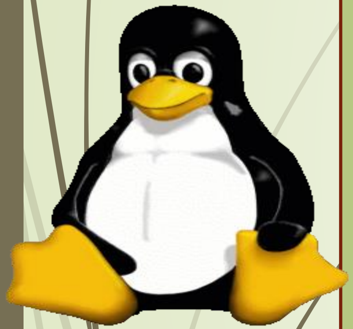

Key Distribution Center (KDC)
(1) $ID_A \| ID_B \| N_1$
Key distribution steps
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s, ID_A])$
(3) $E(K_b, [K_s \| ID_A])$
Initiator A
Responder B
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps

Key Distribution Scenario

- Assume that A wishes to establish a logical connection with B and requires a one-time session key to protect the data transmitted over the connection.
- A has a master key k_a known only to itself and the KDC
- Similarly, B shares the master key k_b with the KDC.



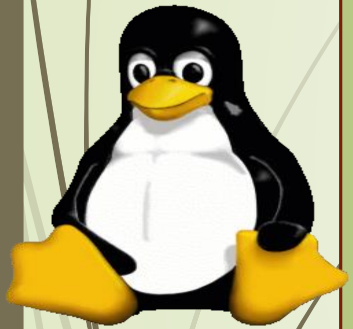
Key Distribution Scenario



Step 1

- A issues a request to the KDC for a session key to protect a logical connection to B.
- The message includes the identity of A and B and a unique identifier N_1 for this transaction referred to as a nonce.
- The nonce may be a timestamp, counter, or a random number. The minimum requirement is that it differs with each request.
- To prevent a masquerade, it should be difficult for an opponent to guess the nonce. Thus a random number is good for a nonce.

Key Distribution Scenario



Step 2

- The KDC responds with a message encrypted using $k_a$thus A is the only one who can successfully read the message and A knows that it originated from the KDC. The message includes two items intended for A,

i. The one-time session key k_s intended for A

ii. The original request message including the nonce to enable A to match this request with the appropriate request

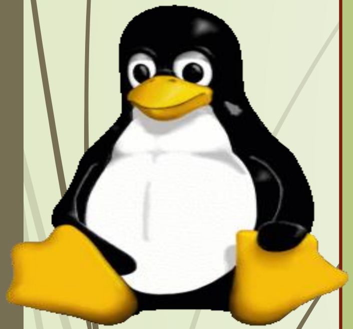
- Thus A can verify that its original request was not altered before reception by the KDC and because of the nonce, that it is not a replay of some previous request.
- The message also includes two items for B

i. The one-time session key k_s to be used for the session

ii. An identifier of A (e.g. its network address), ID_A

This two are encrypted with k_b and are sent to B to establish a connection and prove A's identity

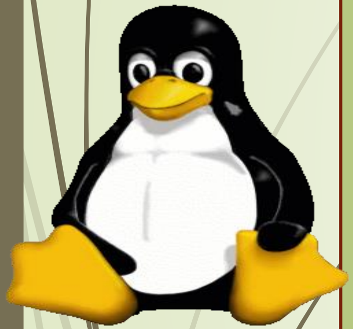
Key Distribution Scenario



Step 3

- A stores the session key for use in upcoming session and forwards to B the information that originated from the KDC namely $E(k_b, [k_s \parallel ID_A])$.
- Because this information is encrypted by k_b, its protected form eavesdropping.
- B now knows the session key k_s, knows the other party is A from ID_A and knows that the information originated from the KDC (because its encrypted by k_b)
- At this point the session key has been securely delivered to A and B and they may begin their protected exchange

Key Distribution Scenario



However two additional steps as desirable

Step 4

- Using the newly minted session key for encryption, B sends a nonce N_2 to A
- Also using K_s, A responds with $f(N_2)$where f is a function that performs some transformations on N_2(e.g. adding one)
- These steps assures B that the original message it received (step 3) was not a replay.

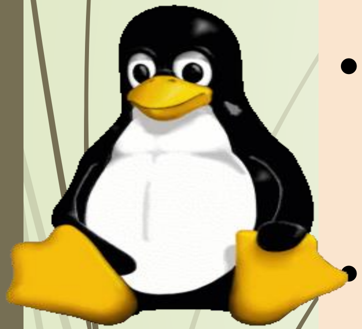
Note: The key distribution only involves step 1 to 3 while step 4 and 5 perform an authentication function.

Major Issues with the KDC

1. Hierarchical key control

- Its not necessary to limit the key distribution function to a single KDC esp. for very large networks.
- As an alternative, a hierarchy of KDC's can be established
- For example, there can be local KDC's each responsible for a small domain of the overall internetwork.
- If two entities in different domains desire a shared key , then the corresponding local KDC's can communicate through the global KDC
- The hierarchical concept can extended to three or even more layers depending on the size of the user population and the geographic scope of the internetwork.
- A hierarchical scheme minimizes the effort involved in master key distribution because most master keys are those shared by the local KDC with its local entities

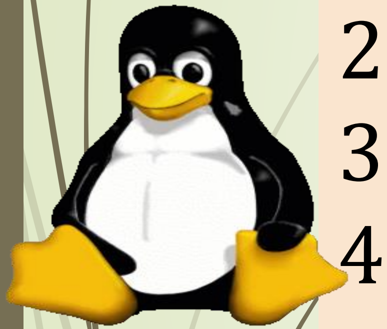
Major Issues with the KDC



2. Session Key Life Time

- The distribution of session keys delays the start of any exchange and places a burden on network capacity
- A security manager must try to balance these competing considerations in determining the lifetime of a particular session key
- For connection oriented protocols, one obvious choice is to use the same session key for the length of time that the connection is open, using a new session key for each session
- If a logical connection has a very long lifetime, it would be prudent to change the session key periodically.
- For connectionless protocol, the is no connection initiation or termination, hence not obvious how often one needs to change the session keys.
  a. The most secure way is to use a new session key for each exchange
  b. Another better strategy is to use a given session key for a certain fixed only or for a certain number of transactions

Techniques of distributing public keys



Public keys can be distributed through one of the below methods

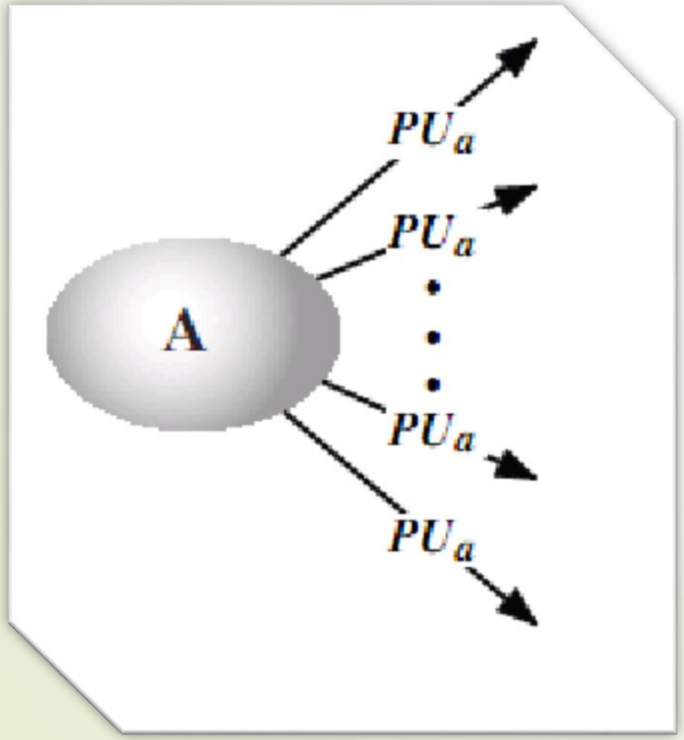
1. Public Announcement
2. Publicly available databases/directories
3. Centralized Distribution: Public key Authority
4. Public key Certificates

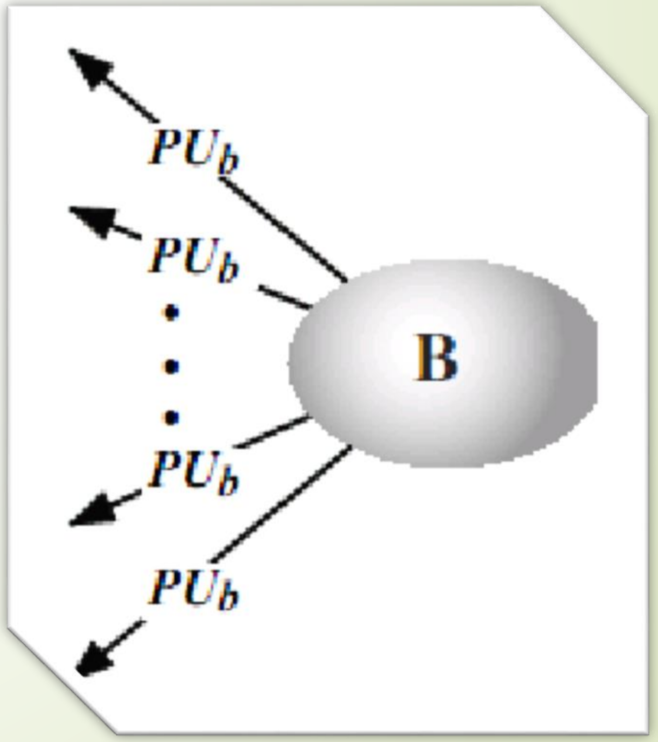
Techniques of distributing public keys

1. Public Announcement

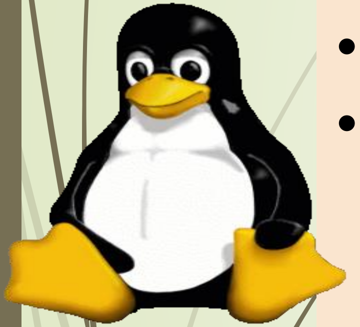
Broadcast your public key to the public

- Via newsgroups, mailing lists, from personal website, etc.
- Major weakness is anyone can easily pretend as yourself (forgery): so attacks are possible





Techniques of distributing public keys



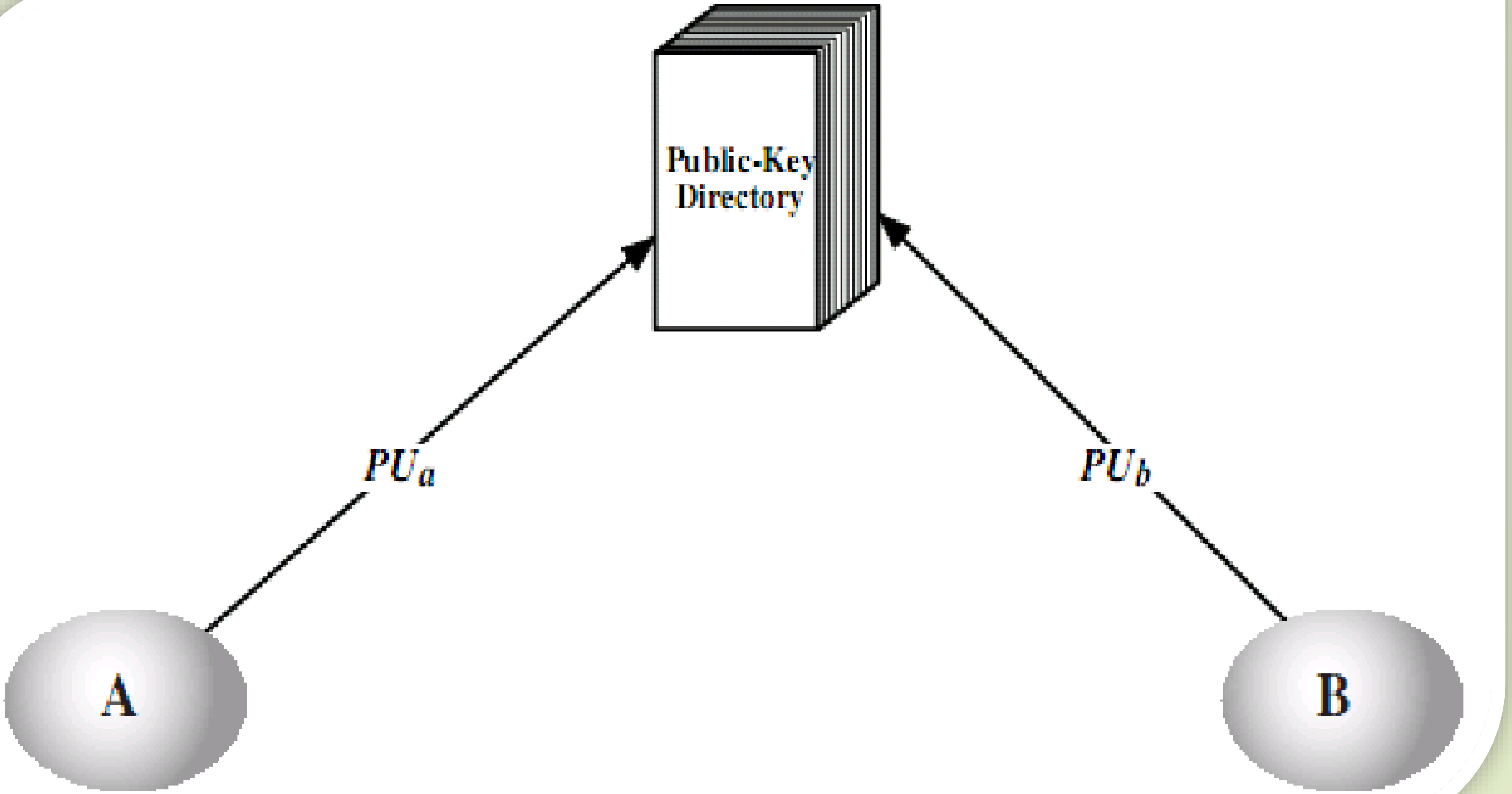
2. Publicly available databases/directories

We can obtain greater security by registering keys with a public directory

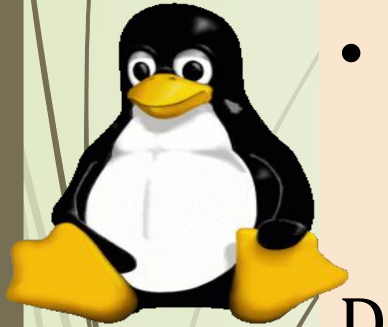
The directory must be trusted with the following properties

- The authority maintains a directory/database with {name, public key} pairs for each participant
- Each participant registers a public key with the directory authority
- A participant may replace the existing key with a new one at any time because the corresponding private key has been compromised in some way
- Participants could also access the directory electronically. For this purpose, secure authenticated communication from the authority to the participant is mandatory
- If administered thoroughly: - good
  - But a proper administration is difficult
    - Need secure mechanisms for registration, update, delete.

Techniques of distributing public keys



Techniques of distributing public keys



3. Centralized Distribution: Public key Authority

- Stronger security for public key distribution can be achieved by providing tighter control over the distribution of public keys from the directory
- It requires users to know the public key for the directory and that they interact with the directory in real-time to obtain any desired public key securely

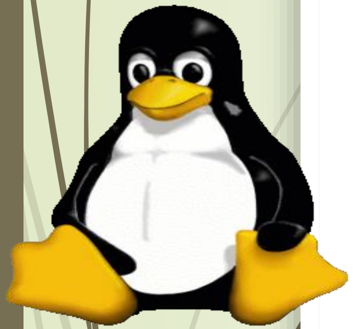
Disadvantages

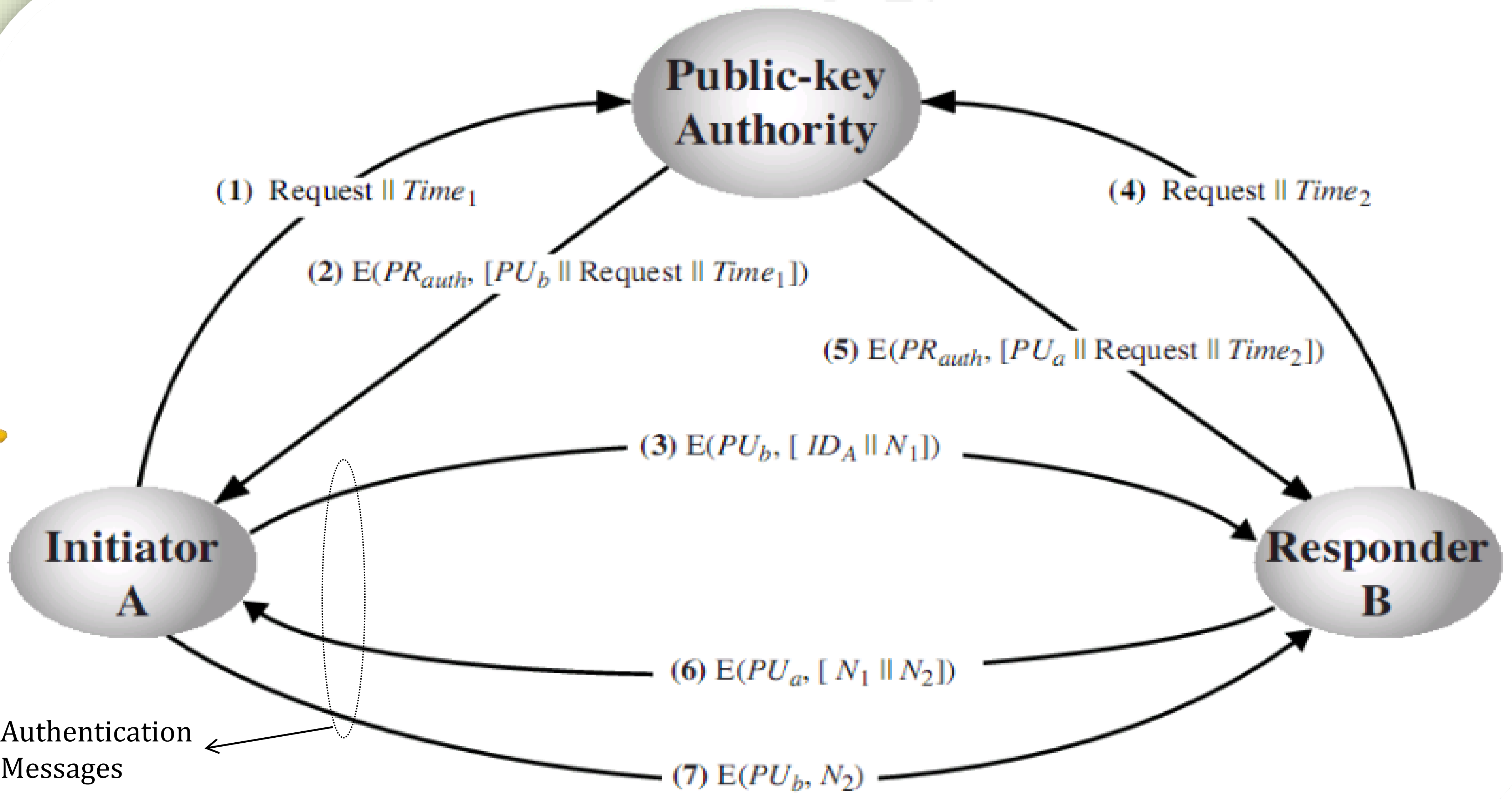
a. Authority is an active entity and may create a performance bottleneck

b. Database should be kept secure to prevent unauthorized modification

- Totally seven messages are required.

Techniques of distributing public keys




Public-key Authority
(1) Request || $Time_1$
(4) Request || $Time_2$
(2) E(PR_{auth}, [PU_b || Request || $Time_1$])
(5) E(PR_{auth}, [PU_a || Request || $Time_2$])
(3) E(PU_b, [ID_A || N_1])
Initiator A
Responder B
(6) E(PU_a, [N_1 || N_2])
(7) E(PU_b, N_2)

Authentication Messages

Techniques of distributing public keys

1. A sends a time stamped message to the public key authority containing a request for the current key of B.
2. The authority responds with a message that is encrypted using the authority's privates key PR_{auth}. Thus A is bale to decrypt the message using the authorities public key. Hence A is assured that the message originated with the authority.

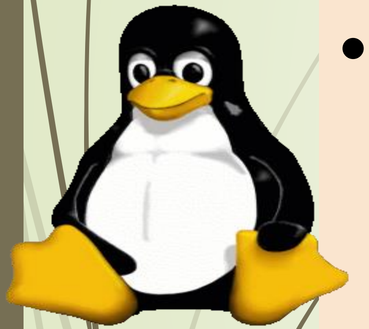
The message includes the following

a. B's public key PU_b which A can use to decrypt messages destined for B
b. The original request, to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority.
c. The original timestamp, so A can determine that this is not an old message from the authority containing a key other than B's current public key.

Techniques of distributing public keys

3. A stores B’s public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1) which is used to identify this transaction uniquely.
4. B retrieves A’s public key from the authority in the same manner as A retrieved B’s public key.
5. At this point, public keys have been securely delivered to A and B and they may begin there protected exchange. However two additional steps are desired.
6. B sends A a message encrypted with $\mathrm{PU_a}$ and containing A’s nonce (N_1) as well as a new nonce generated by B (N_2) . Because only B could have decrypted message (3), the presence of (N_1) in message (6) assures A that the correspondent is B.
7. A returns (N_2) , encrypted using B’s public key, to assure B that its correspondent is A.

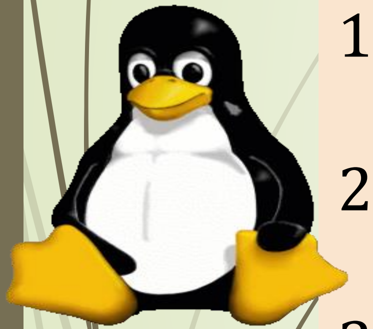
Techniques of distributing public keys



4. Public key Certificates

- PKC’s can be used to exchange keys without contacting a public key authority
- A certificate binds an **identity** to **public keys** with all contents signed by a trustee public-key or Certificate Authority(CA)
- This can be verified by anyone who knows the public-key authorities public-key

Techniques of distributing public keys



4. Public key Certificates

A participant can also convey its key information to another by transmitting its certificate

Other participants can verify that the certificate was created by the authority,

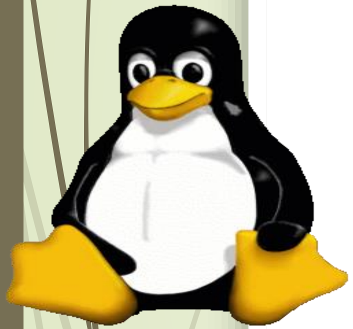
The following requirements can be placed on this scheme

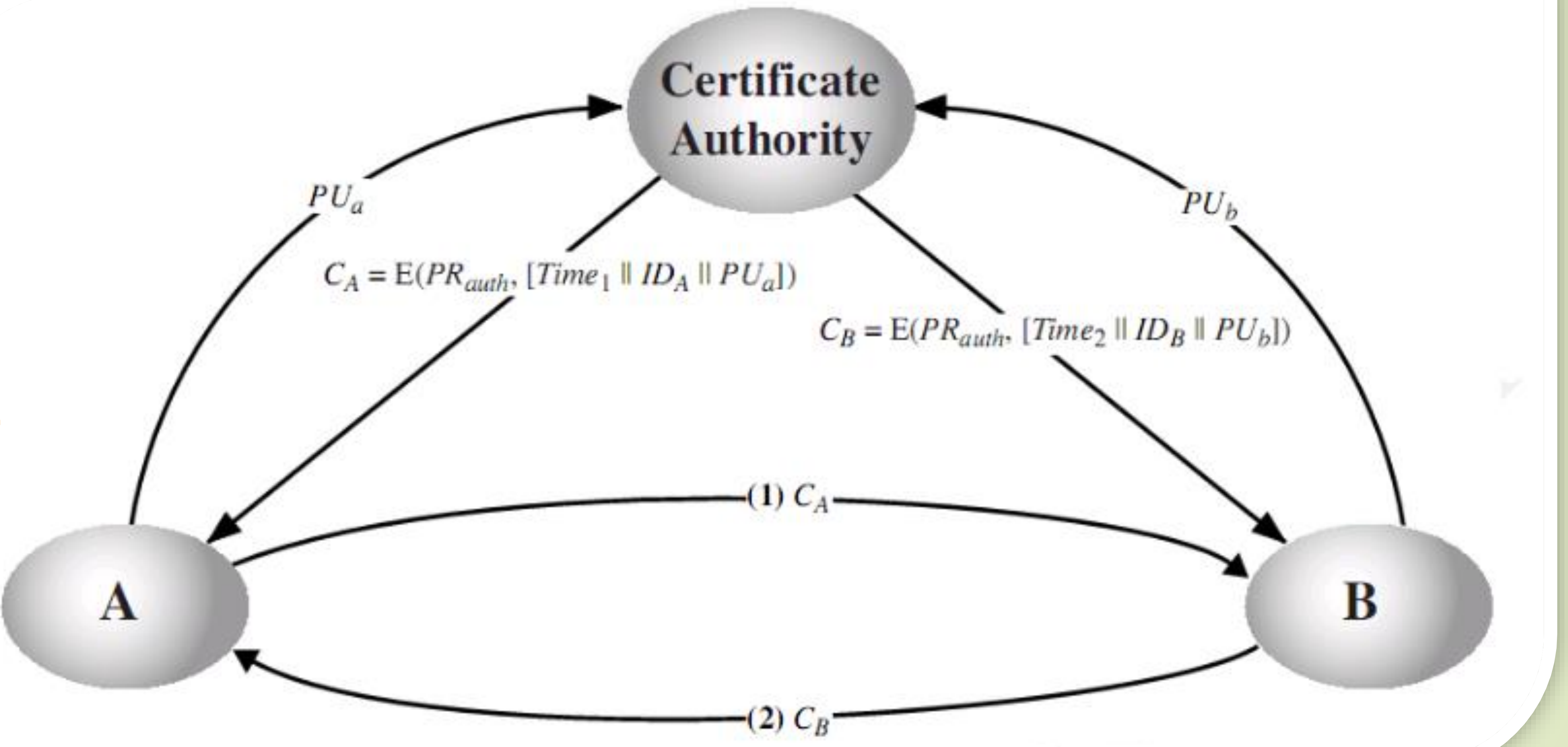
1. Any participant can read a certificate to determine the name and public key of the certificate's owner
2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit
3. Only the certificate authority can create and update certificates
4. Any participant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

One such scheme has become universally accepted for formatting public-key certificates: the X.509 standards

X.509 certificates are used in most network security applications, including IP security, secure socket layer (SSL), secure electronic transactions (SET).

Techniques of distributing public keys




Certificate Authority
PU_a
PU_b
$C_A = E(PR_{auth}, [Time_1 \| ID_A \| PU_a])$
$C_B = E(PR_{auth}, [Time_2 \| ID_B \| PU_b])$
(1) C_A
(2) C_B
A
B

STOP