Cryptography and Network Security Chapter 14

Fifth Edition

by William Stallings

Lecture slides by Lawrie Brown

Chapter 14 – Key Management and Distribution

No Singhalese, whether man or woman, would venture out of the house without a bunch of keys in his hand, for without such a talisman he would fear that some devil might take advantage of his weak state to slip into his body.

—The Golden Bough, Sir James George Frazer

Key Management and Distribution

- topics of cryptographic key management / key distribution are complex
  - cryptographic, protocol, & management issues
- **symmetric schemes** require both parties to share a common secret key
- **public key schemes** require parties to acquire valid public keys
- have concerns with doing both

Key Distribution

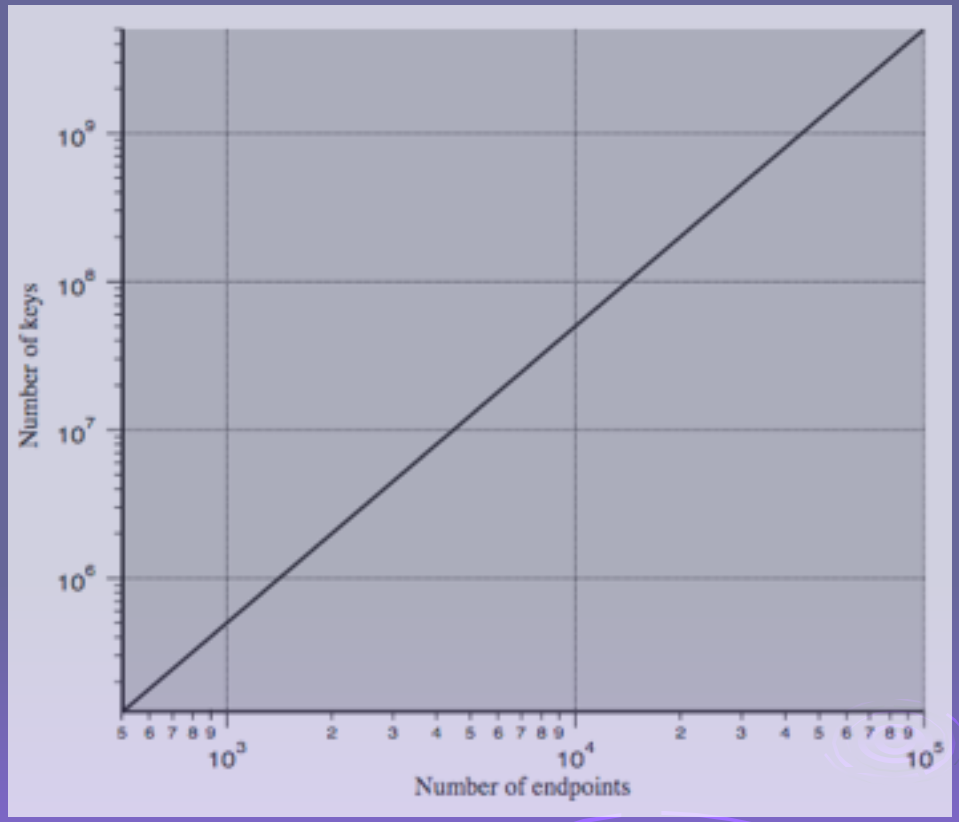
- **symmetric schemes** require both parties to share a common secret key
- issue is **how to securely distribute this key** whilst **protecting** it from others
- **frequent key changes** can be desirable
- often secure system failure due to a break in the key distribution scheme

Key Distribution

- Given parties A and B have various **key distribution** alternatives:
  - *A* can select key and physically deliver to *B*.
  - A third party can select the key and physically deliver it to *A* and *B*.
  - If *A* and *B* have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
  - If *A* and *B* each has an encrypted connection to a third party *C*, *C* can deliver a key on the encrypted links to *A* and *B*.

Key Distribution Task

- If end-to-end encryption is done at a network or IP level, then a key is needed for each pair of hosts.
- If there are N hosts, the number of required keys is [N(N-1)/2].



Key Hierarchy

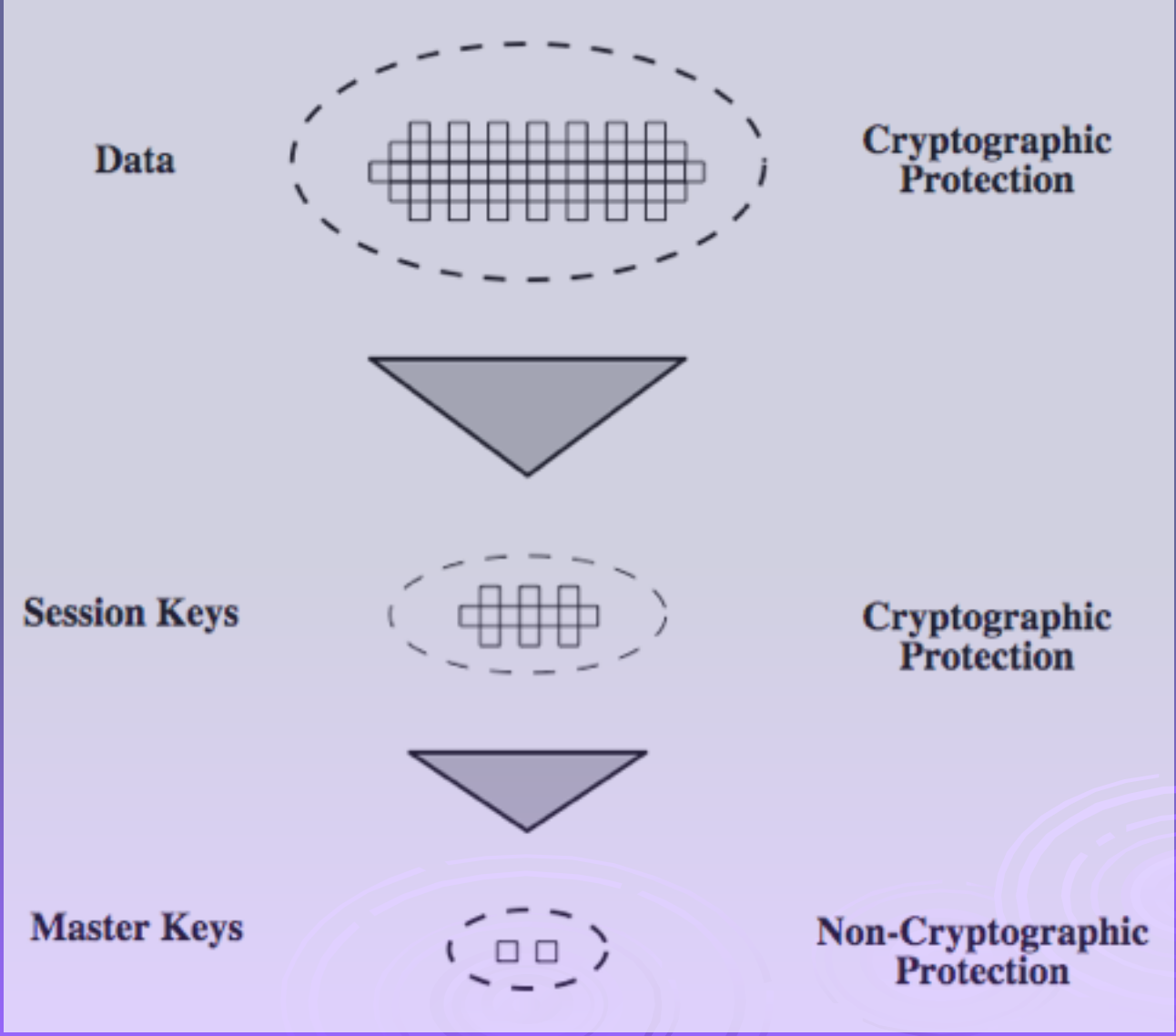
- typically have a hierarchy of keys
- session key
  - temporary key
  - used for encryption of data between users
  - for one logical session then discarded
- master key
  - used to encrypt session keys
  - shared by user & key distribution center

Key Hierarchy

- The use of a key distribution center is based on the use of a hierarchy of keys.

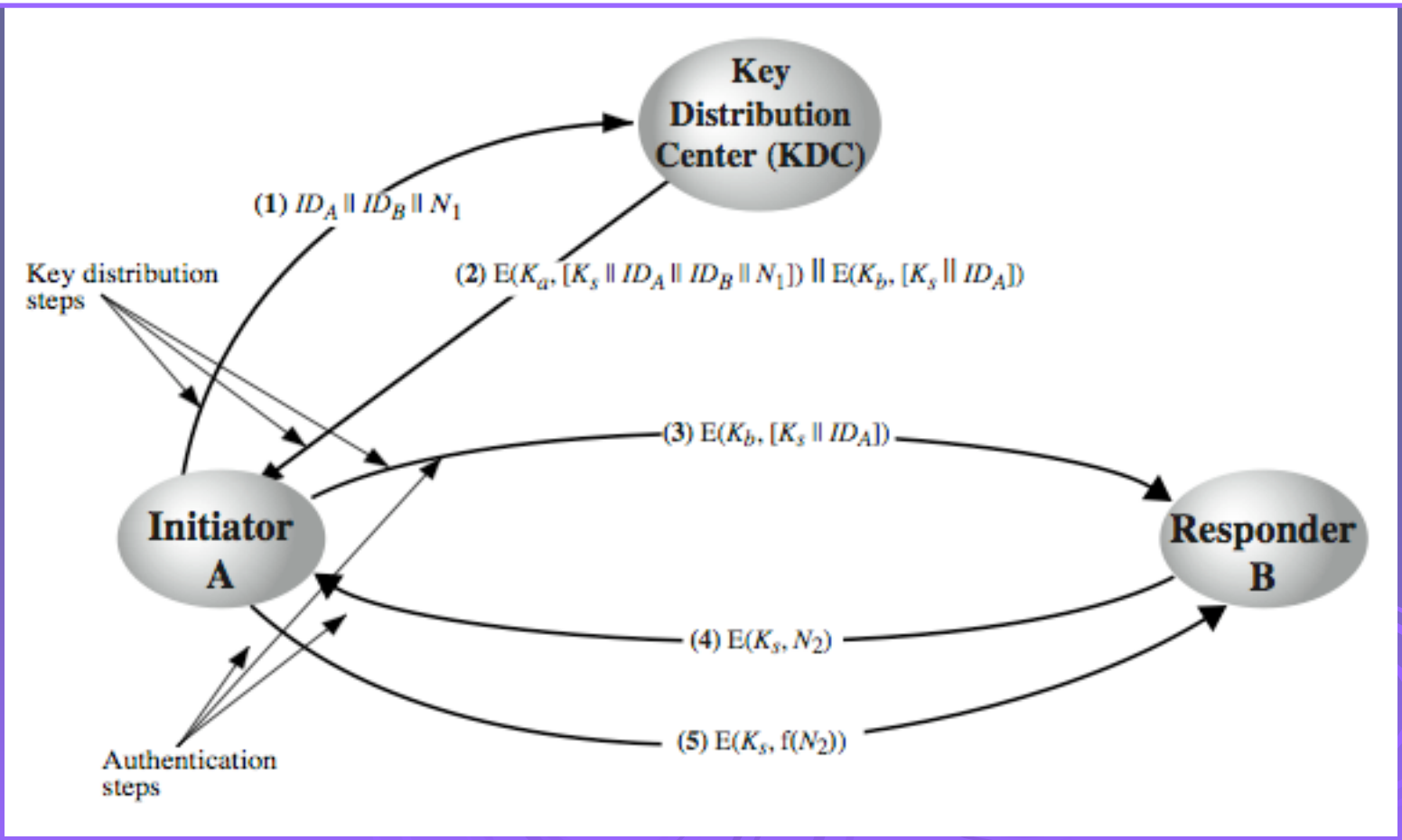
- session key
  - temporary key
  - used for encryption of data between users for one logical session then discarded
- master key
  - used to encrypt session keys
  - shared by user & key distribution centre

Key Hierarchy



Key Distribution Scenario

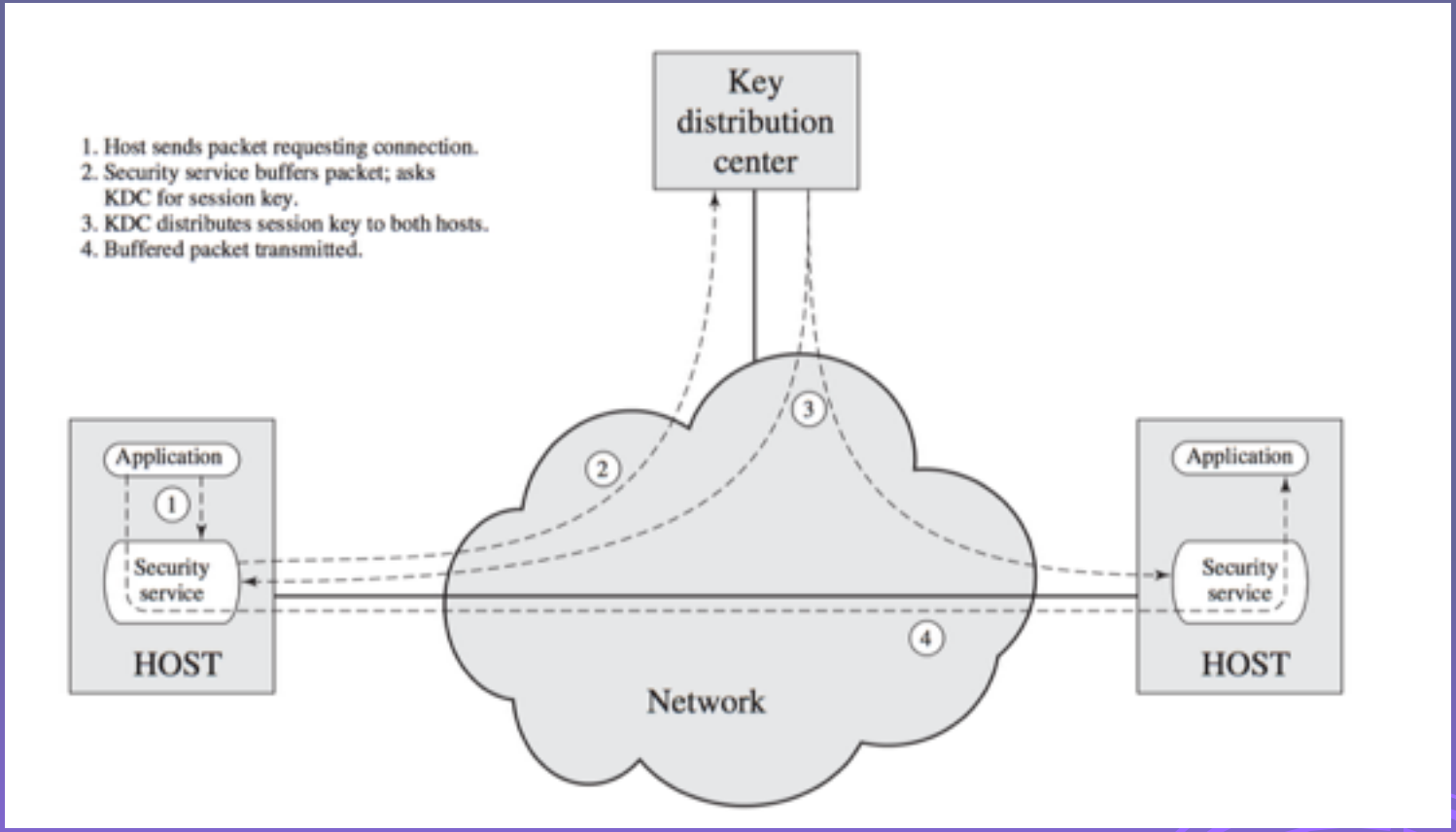
The scenario assumes that each user shares a unique master key with the key distribution center (KDC).



Key Distribution Issues

- Hierarchies of KDC's : required for large networks, but must trust each other
- Session key lifetimes: should be limited for greater security
- use of automatic key distribution on behalf of users, but must trust system
- use of decentralized key distribution
- controlling key usage

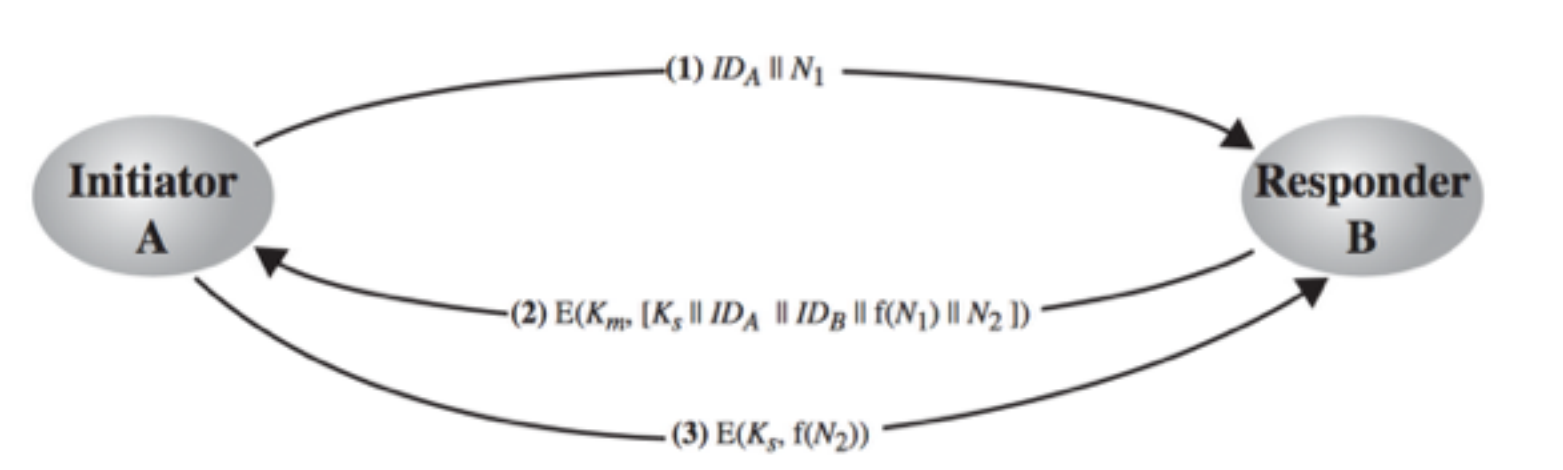
A Transparent Key Control Scheme



Benefits

- The automated key distribution approach provides the flexibility and dynamic characteristics needed to allow a number of terminal users to access a number of hosts and for the hosts to exchange data with each other.

decentralized key distribution



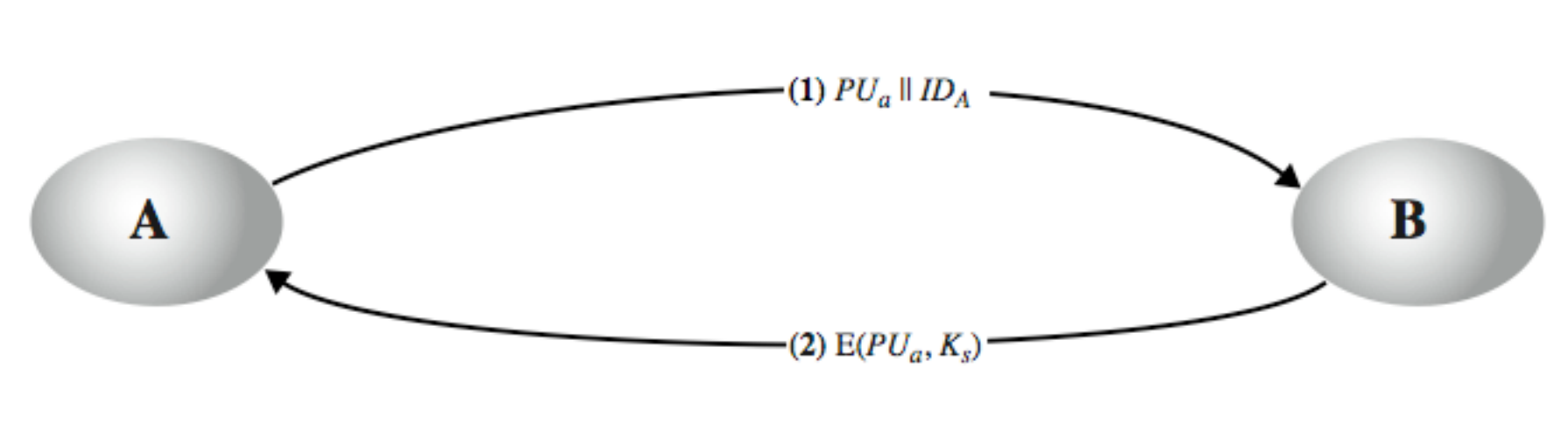
- each end system be able to communicate in a secure manner with all potential partner end systems for purposes of session key distribution.
- each node must maintain at most (n - 1) master keys, as many session keys as required may be generated and used

Symmetric Key Distribution Using Public Keys

- public key cryptosystems are inefficient
  - so almost never use for direct data encryption
  - rather use to encrypt secret keys for distribution

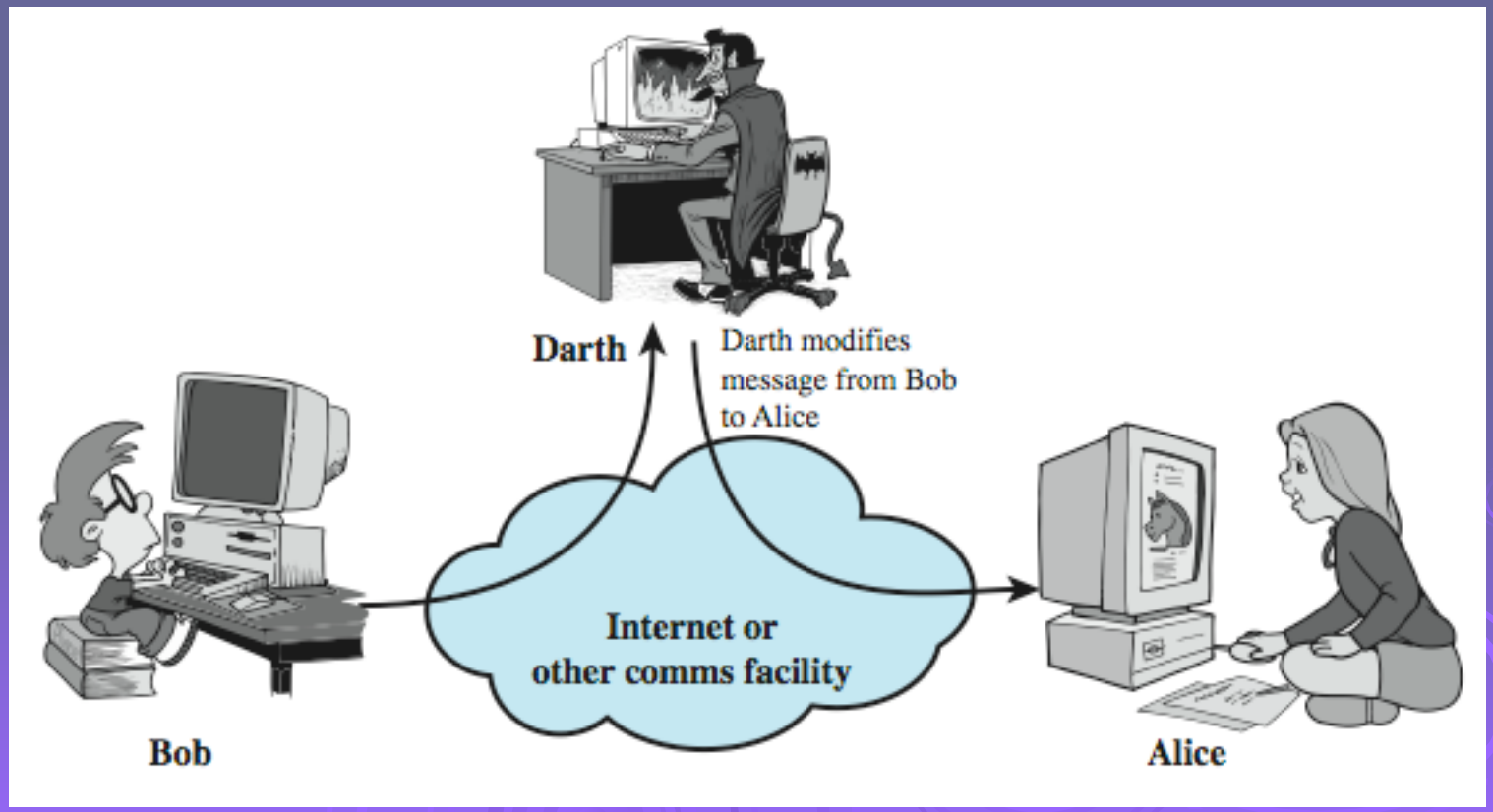
Simple Secret Key Distribution

- Merkle proposed this very simple scheme
  - allows secure communications
  - no keys before/after exist

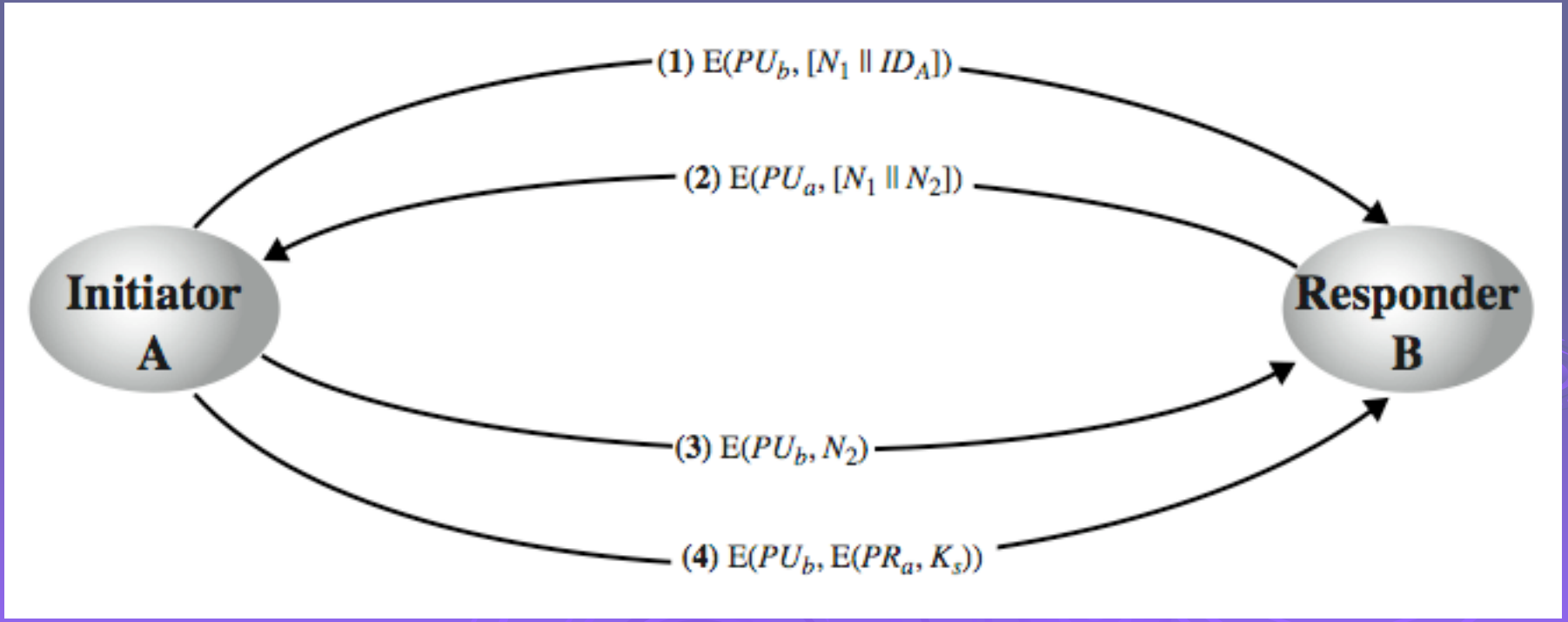


Man-in-the-Middle Attack

- this very simple scheme is vulnerable to an active man-in-the-middle attack



Secret Key Distribution with Confidentiality and Authentication



Hybrid Key Distribution

- retain use of private-key KDC
- shares secret master key with each user
- distributes session key using master key
- public-key used to distribute master keys
  - especially useful with widely distributed users
- rationale
  - performance
  - backward compatibility

Distribution of Public Keys

- can be considered as using one of:
  - public announcement
  - publicly available directory
  - public-key authority
  - public-key certificates

Public Announcement

- users distribute public keys to recipients or broadcast to community at large
  - eg. append PGP keys to email messages or post to news groups or email list
- major weakness is forgery
  - anyone can create a key claiming to be someone else and broadcast it
  - until forgery is discovered can masquerade as claimed user

Publicly Available Directory

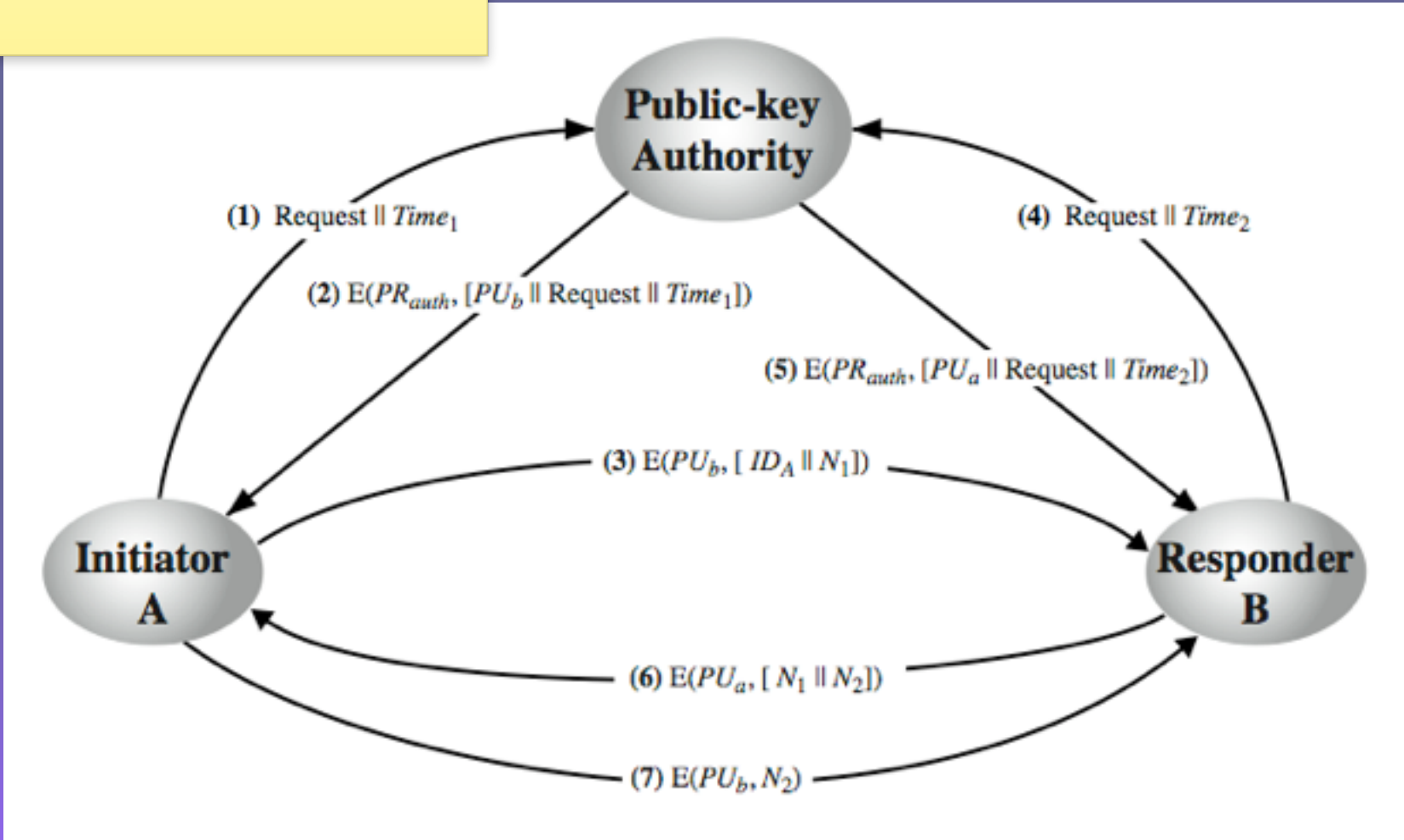
- can obtain greater security by registering keys with a public directory
- directory must be trusted with properties:
  - contains {name,public-key} entries
  - participants register securely with directory
  - participants can replace key at any time
  - directory is periodically published
  - directory can be accessed electronically
- still vulnerable to tampering or forgery

Public-Key Authority

- improve security by tightening control over distribution of keys from directory
- has properties of directory
- and requires users to know public key for the directory
- then users interact with directory to obtain any desired public key securely
  - does require real-time access to directory when keys are needed
  - may be vulnerable to tampering

include a timestamp to ensure exact timeliness, though this requires clock synchronization between organizations
DFT('YYYY-MM-DD-HH.MM.SS.UUUUUU')

c-Key Authority



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, PR_{auth}.

   - B's public key, PU_b, which A can use to encrypt messages destined for B

   - The original request used 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

   - The original timestamp given so A can determine that this is not an old message from the authority containing a key other than B's current public key

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, 5. B retrieves A's public key from the authority in the same manner as A retrieved B's public key.

6. B sends a message to A encrypted with 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, which is encrypted using B's public key, to assure B that its correspondent is A.

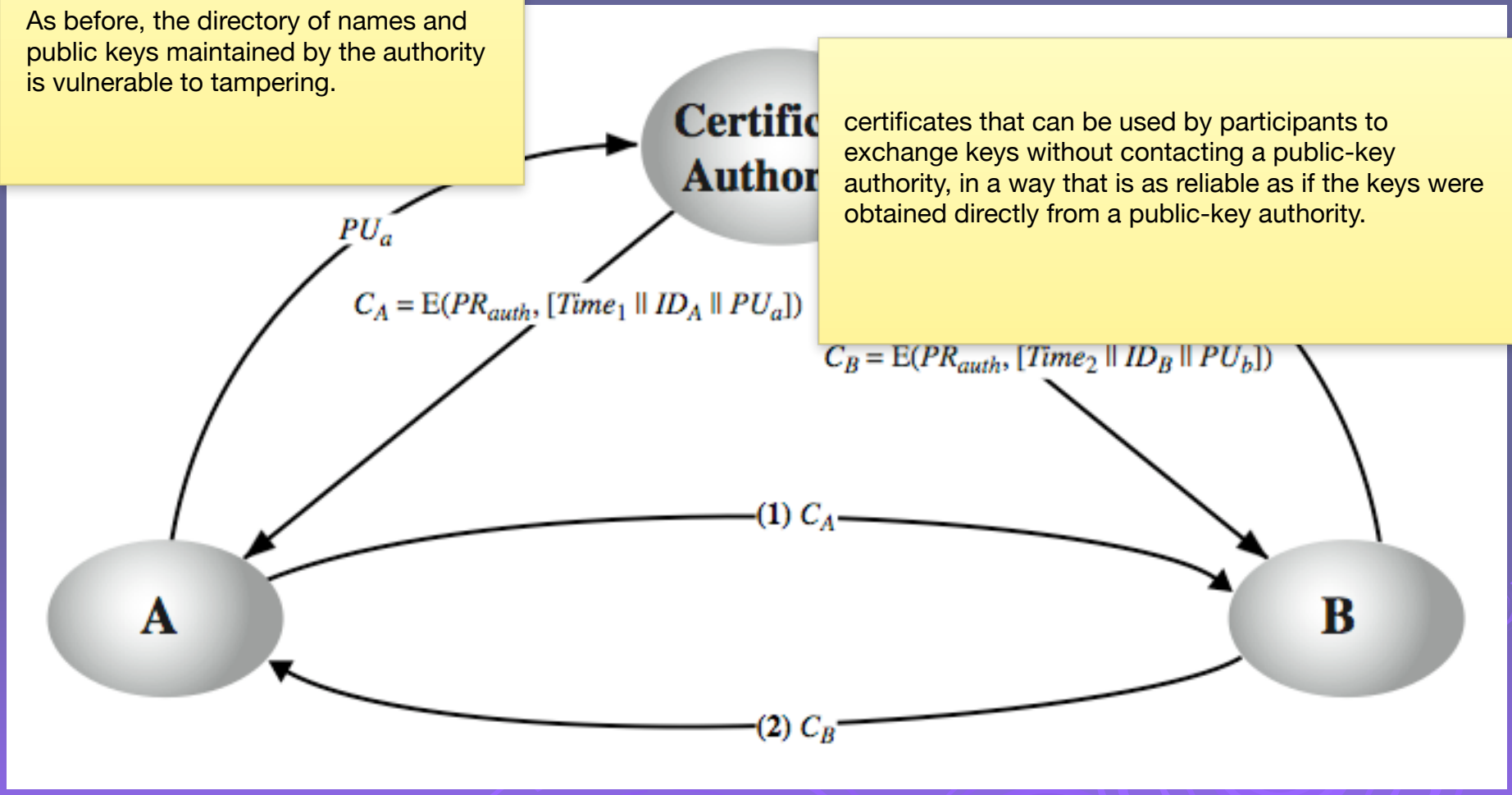
Public-Key Certificates

- certificates allow key exchange without real-time access to public-key authority
- a certificate binds **identity** to **public key**
  - usually with other info such as period of validity, rights of use etc
- with all contents **signed** by a trusted Public-Key or Certificate Authority (CA)
- can be verified by anyone who knows the public-key authorities public-key

-Key Certificates

for a user must appeal to the authority for a public key for every other user that it wishes to contact.
As before, the directory of names and public keys maintained by the authority is vulnerable to tampering.

certificates that can be used by participants to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority.

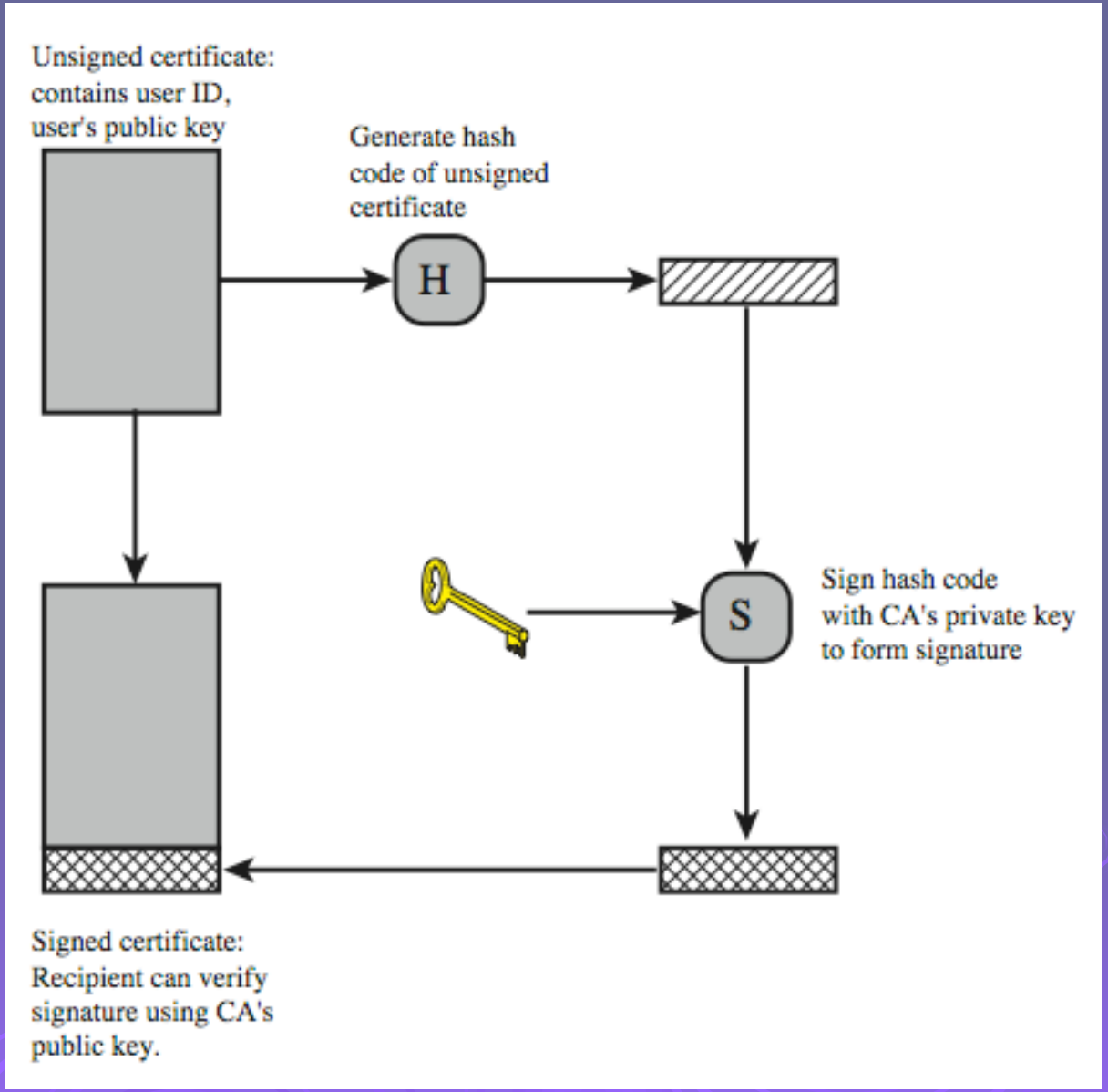


X.509 Authentication

Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.

- part of CCITT X.500 that defines service standards
  - distributed servers maintaining user info database
- defines framework for authentication services
  - directory may store public-key certificates
  - with public key of user signed by certification authority
- also defines authentication protocols
- uses public-key crypto & digital signatures
  - algorithms not standardised, but RSA recommended
- X.509 certificates are widely used
  - have 3 versions

X.509 Certificate Use



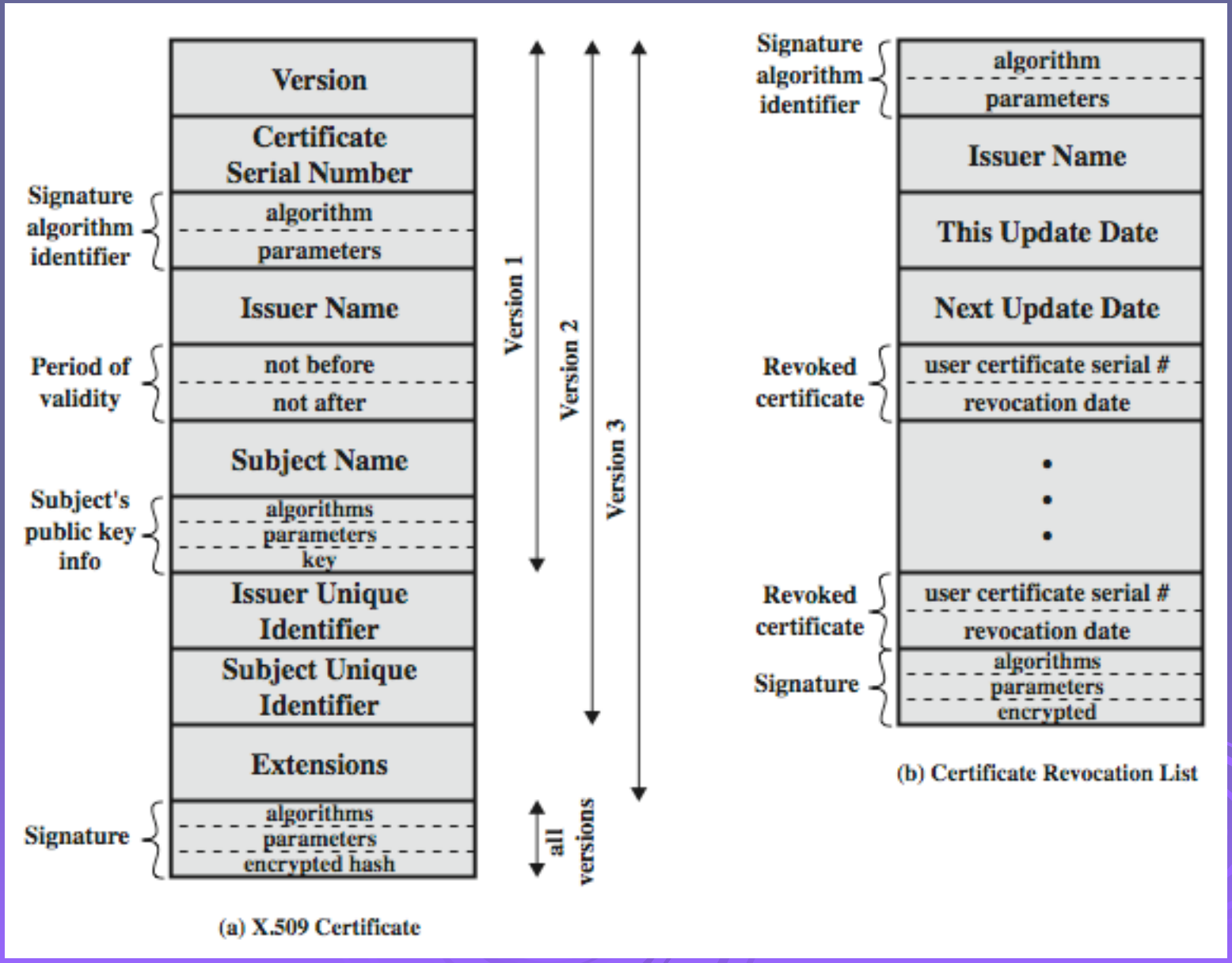
X.509 Certificate

A certificate contains a public key. The certificate, in addition to the pu key, contains additional information as issuer, what it's supposed to be for, and any other type of metadata. Typically a certificate is itself signed a private key, that verifies its authen

- issued by a Certification Authority (CA), co
  - version V (1, 2, or 3)
  - serial number **SN** (unique within CA) identifying
  - signature algorithm identifier ***AI***
  - issuer X.500 name ***(CA)***
  - period of validity ***TA*** (from - to dates)
  - subject X.500 name ***A*** (name of owner)
  - subject public-key info Ap (algorithm, parameters, key) ***Ap***
  - issuer unique identifier (v2+) ***UCA***
  - subject unique identifier (v2+) ***UA***
  - extension fields (v3)
  - signature (of hash of all fields in certificate)
- notation `CA<<A>>` denotes certificate for A signed by CA

CA{V,SN,AI,CA,UCA,A,UA,Ap,TA}

X.509 Certificates

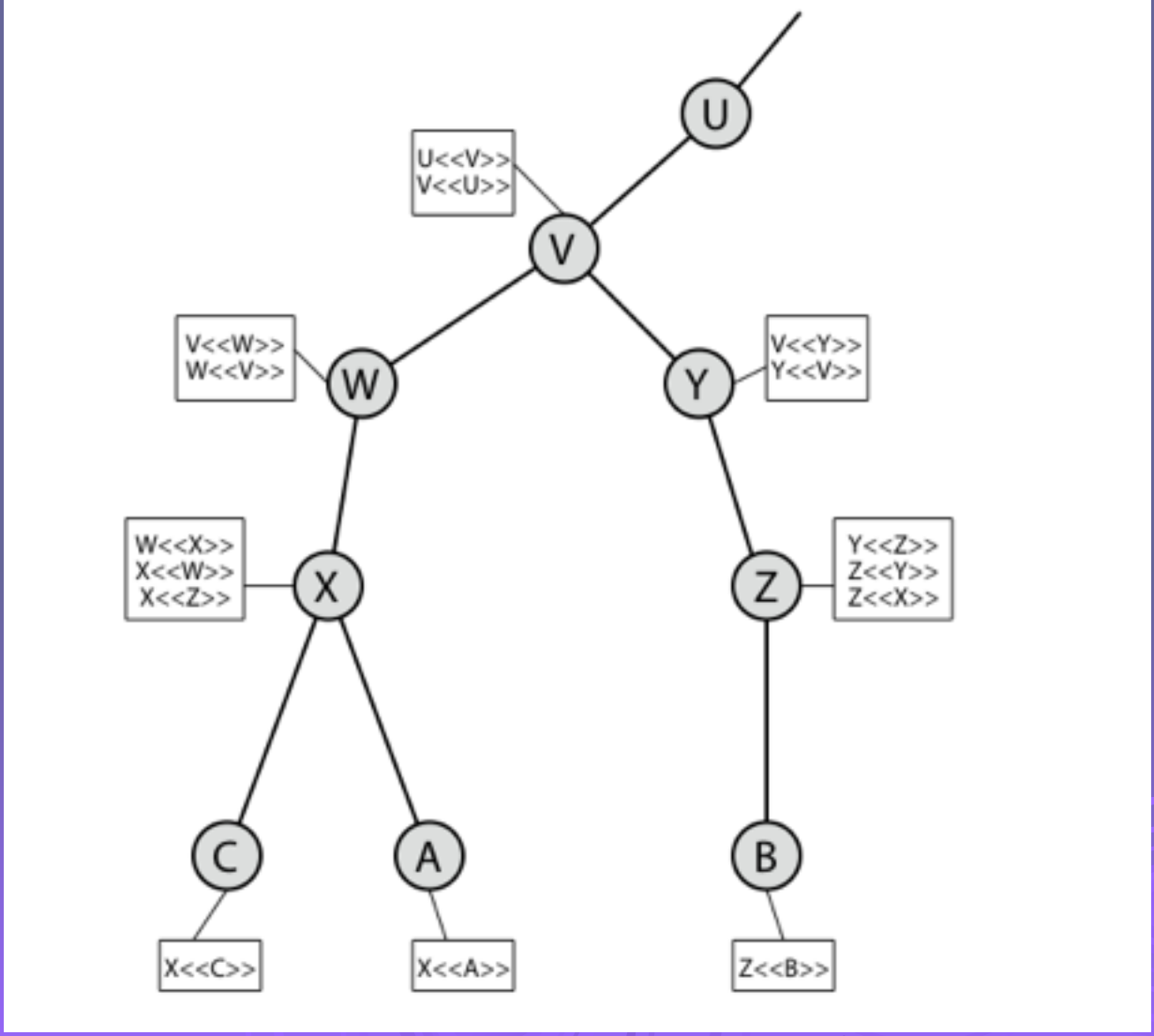


Obtaining a Certificate

- any user with access to CA can get any certificate from it
- only the CA can modify a certificate
- because cannot be forged, certificates can be placed in a public directory

- if both users share a common CA then they are assumed to know its public key
- otherwise CA's must form a hierarchy
- use certificates linking members of hierarchy to validate other CA's
  - each CA has certificates for clients (forward) and parent (backward)
- each client trusts parents certificates
- enable verification of any certificate from one CA by users of all other CAs in hierarchy

CA Hierarchy Use



In this example, user A can acquire the following certificates from the directory to establish a certification path to B:

$$X \ll W \gg \; W \ll V \gg \; V \ll Y \gg \; Y \ll Z \gg \; Z \ll B \gg$$

B require A's public key which can be obtained from the following certification path:

$$Z \ll Y \gg \; Y \ll V \gg \; V \ll W \gg \; W \ll X \gg \; X \ll A \gg$$

Certificate Revocation

- certificates have a period of validity
- may need to revoke before expiry, eg:
  1. user's private key is compromised
  2. user is no longer certified by this CA
  3. CA's certificate is compromised
- CA's maintain list of revoked certificates
  - the Certificate Revocation List (CRL)
- users should check certificates with CA's CRL

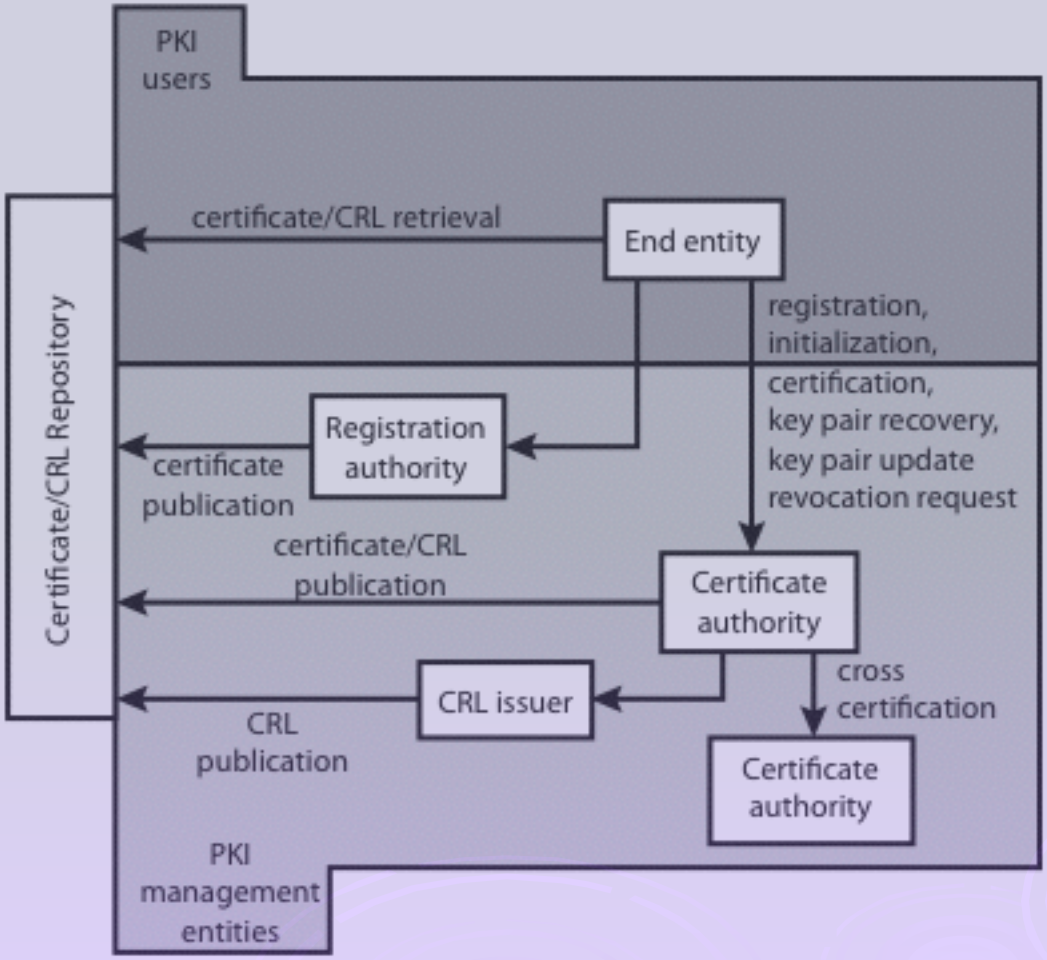
X.509 Version 3

- has been recognised that additional information is needed in a certificate
  - email/URL, policy details, usage constraints
- rather than explicitly naming new fields defined a general extension method
- extensions consist of:
  - extension identifier
  - criticality indicator
  - extension value

Certificate Extensions

- key and policy information
  - convey info about subject & issuer keys, plus indicators of certificate policy
- certificate subject and issuer attributes
  - support alternative names, in alternative formats for certificate subject and/or issuer
- certificate path constraints
  - allow constraints on use of certificates by other CA's

Public Key Infrastructure



PKIX Management

- functions:
  - registration
  - initialization
  - certification
  - key pair recovery
  - key pair update
  - revocation request
  - cross certification
- protocols: CMP, CMC

Summary

- have considered:
  - symmetric key distribution using symmetric encryption
  - symmetric key distribution using public-key encryption
  - distribution of public keys
    - announcement, directory, authrority, CA
  - X.509 authentication and certificates
  - public key infrastructure (PKIX)