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

Road Map

- 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)

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
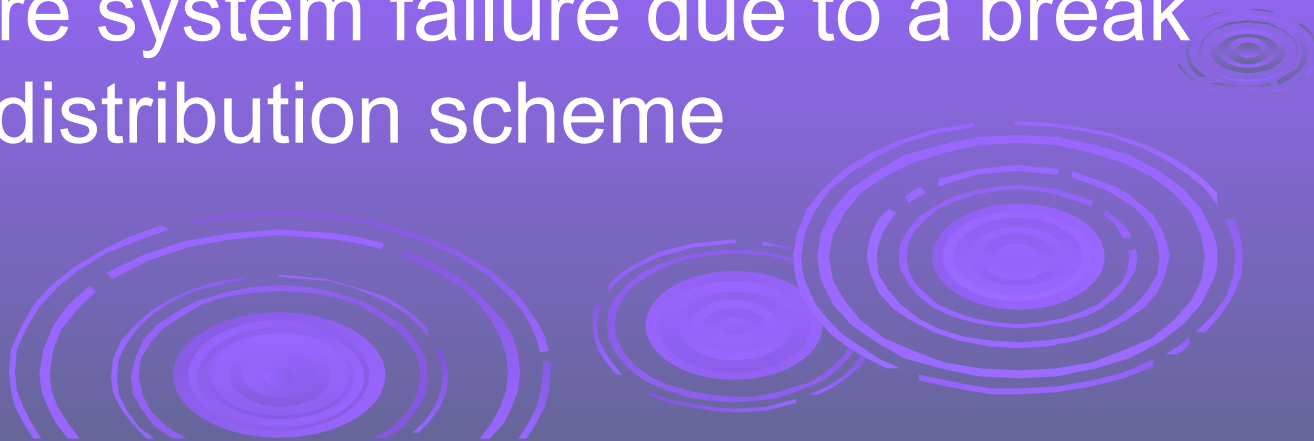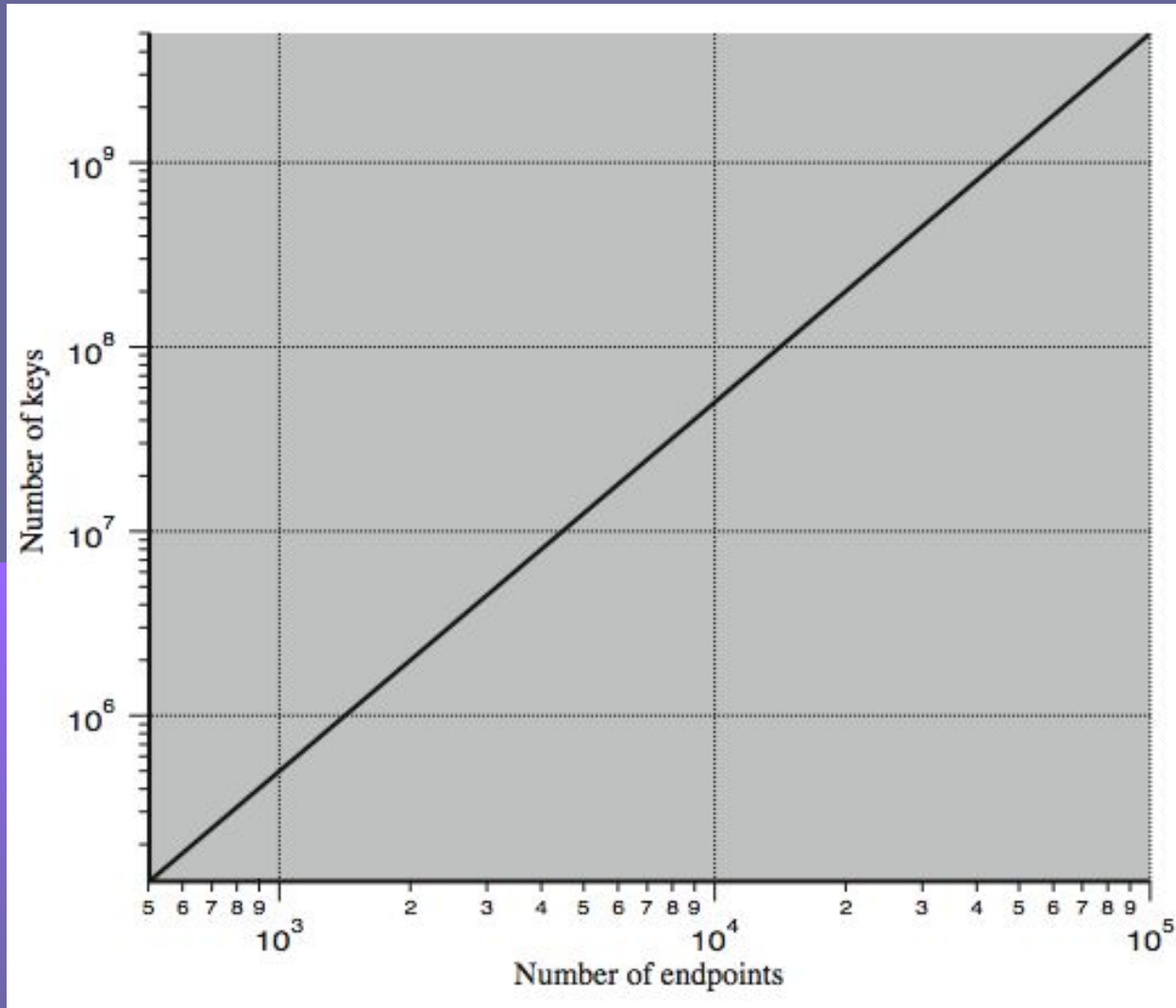
Key Distribution

given parties A and B have various **key distribution** alternatives:

1. A can select key and physically deliver to B
2. third party can select & deliver key to A & B
3. if A & B have communicated previously can use previous key to encrypt a new key
4. if A & B have secure communications with a third party C, C can relay key between A & B
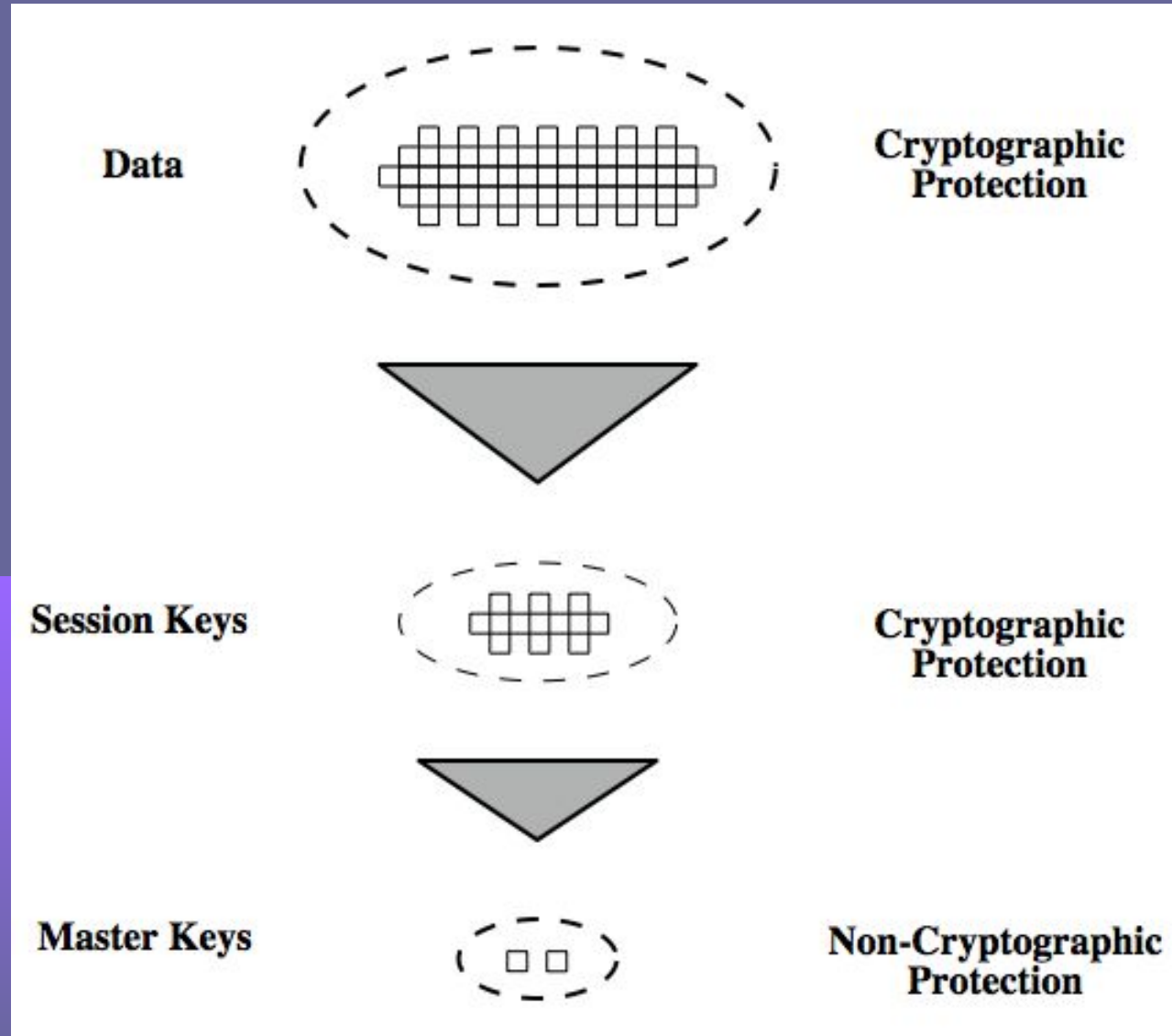
Key Distribution Task
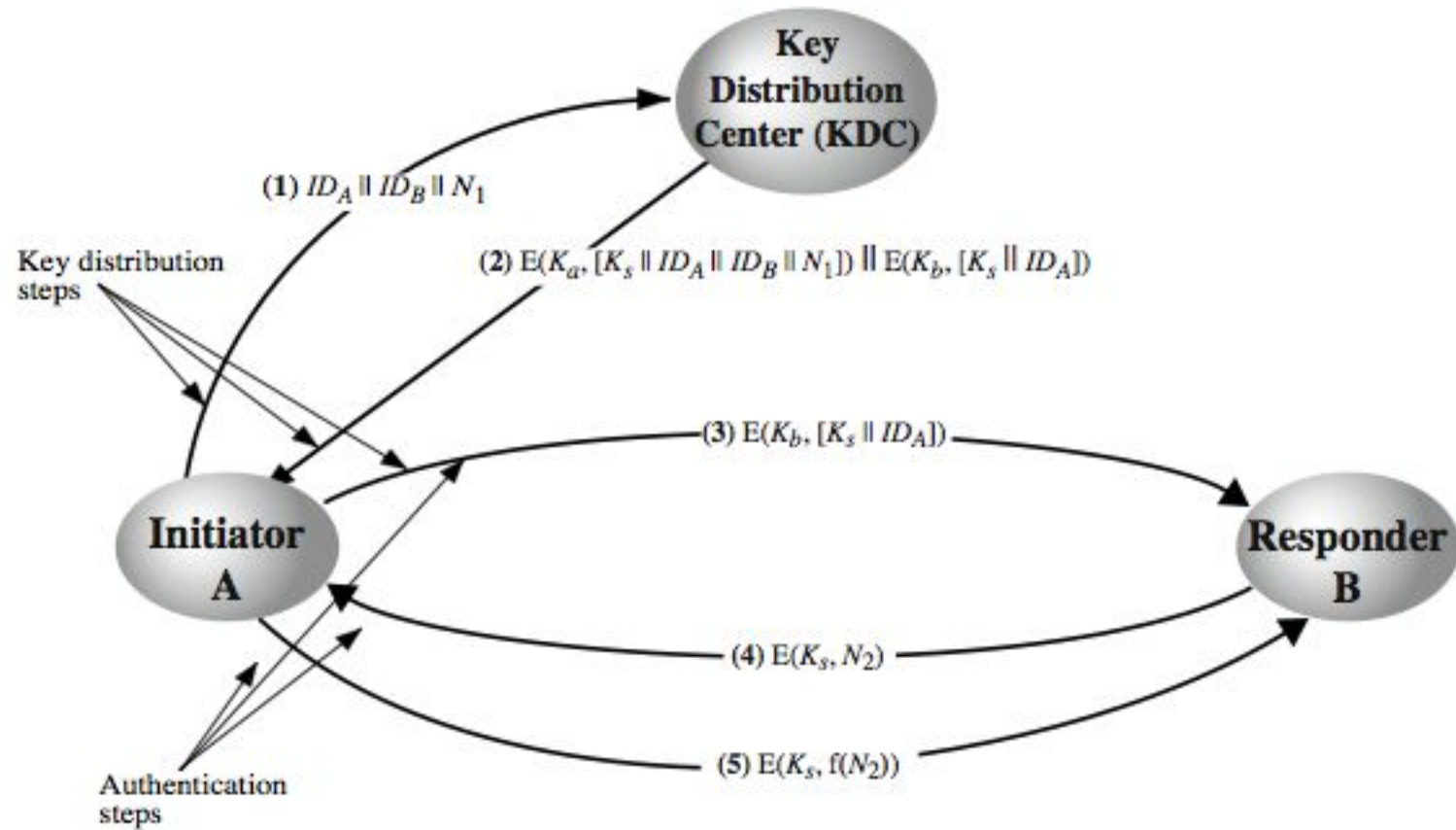
Key Hierarchy

- 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
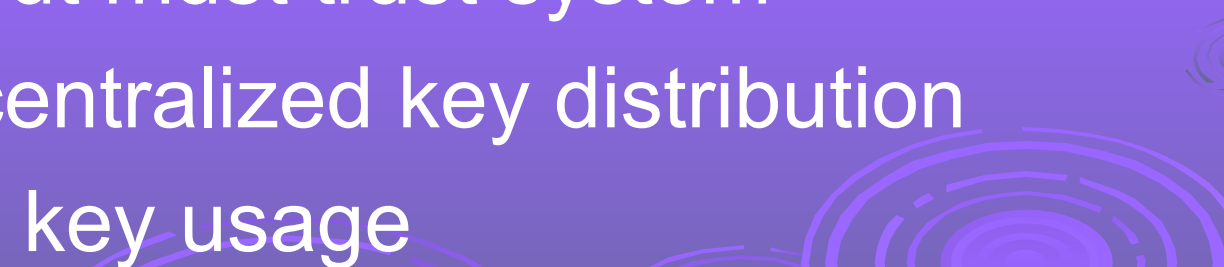 - shared by user & key distribution center

Key Hierarchy

Data — Cryptographic Protection

Session Keys — Cryptographic Protection

Master Keys — Non-Cryptographic Protection

Key Distribution Scenario

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
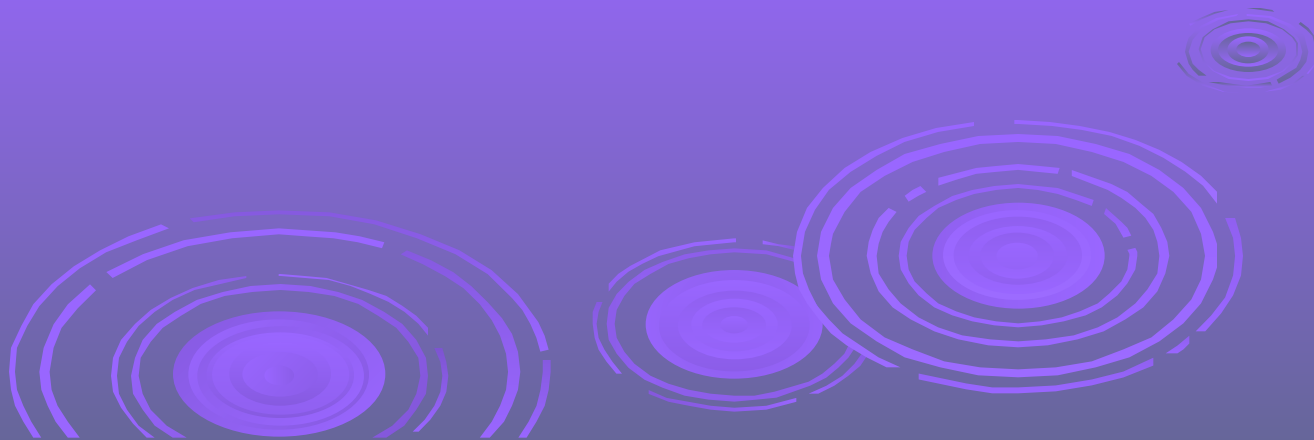- controlling key usage

Road Map

- 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)

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

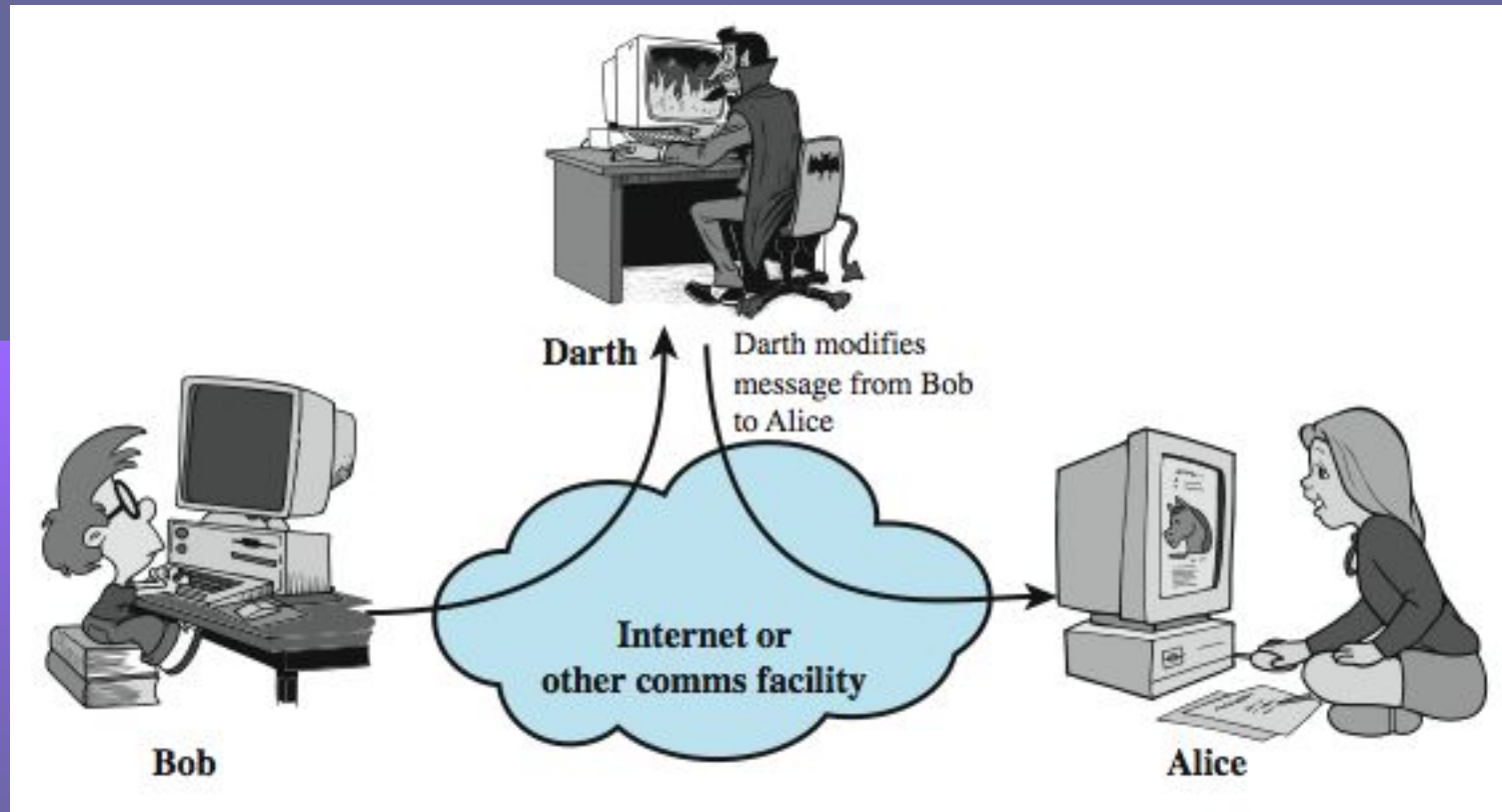# Simple Secret Key Distribution

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

$$(1)\ PU_a \parallel ID_A$$

A

B

$$(2)\ E(PU_a, K_s)$$
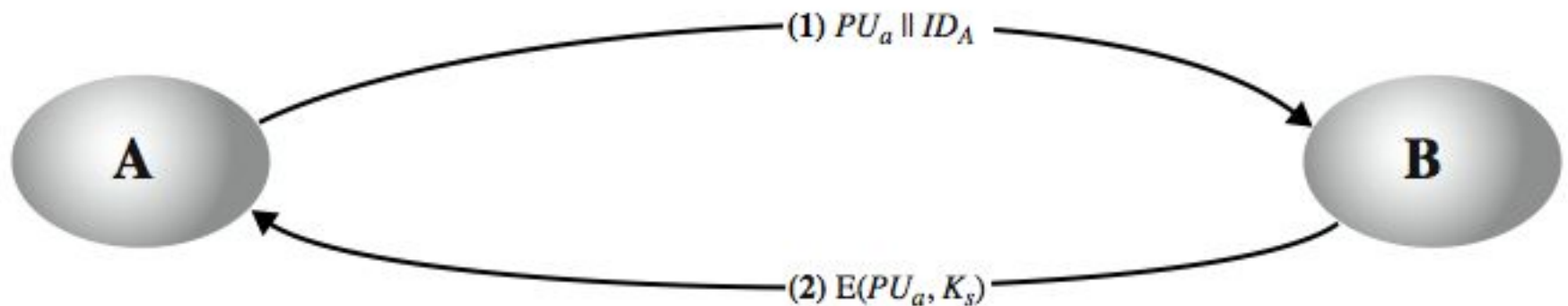
Man-in-the-Middle Attack

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

Darth

Darth modifies message from Bob to Alice

Internet or other comms facility

Bob

Alice

Secret Key Distribution with Confidentiality and Authentication

(1) $E(PU_b, [N_1 \| ID_A])$

(2) $E(PU_a, [N_1 \| N_2])$

(3) $E(PU_b, N_2)$

(4) $E(PU_b, E(PR_a, K_s))$

Initiator A

**Responder B**
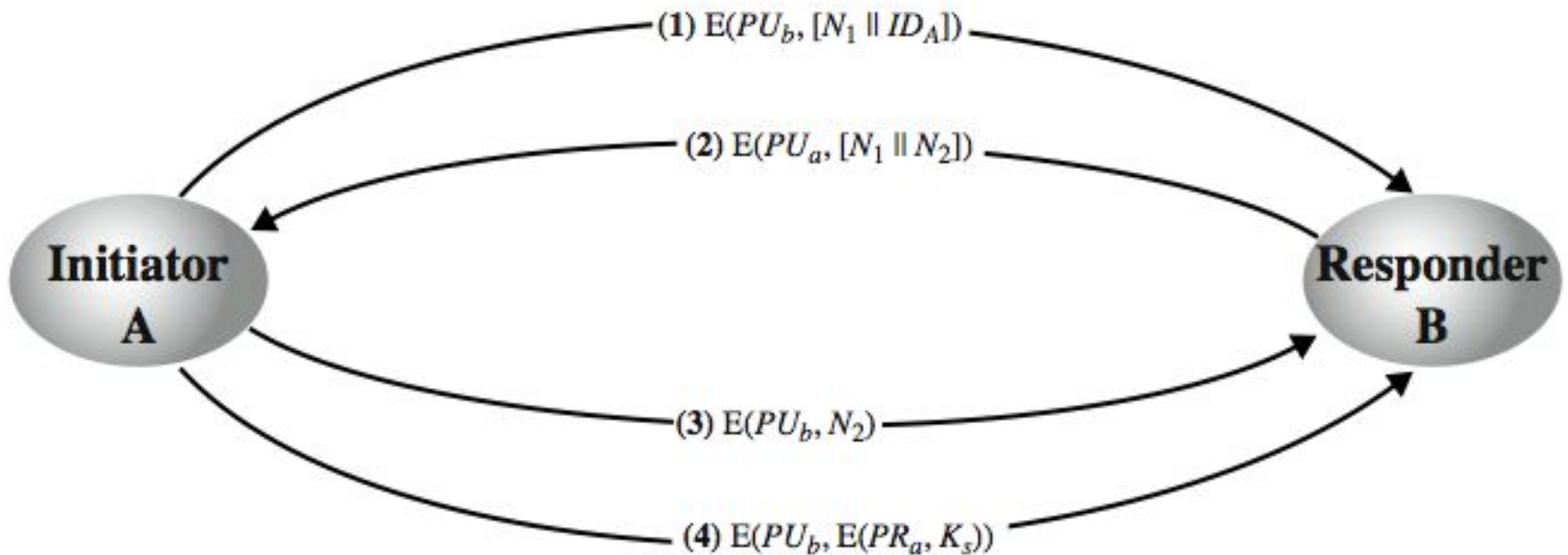
Hybrid Key Distribution

- 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
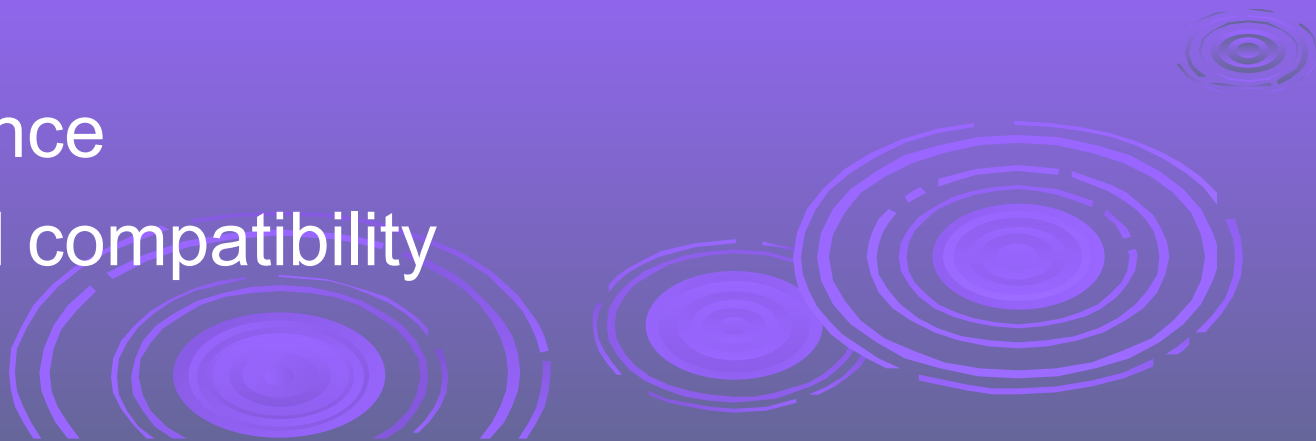
Road Map

- symmetric key distribution using symmetric encryption
- symmetric key distribution using public-key encryption
- distribution of public keys
  - announcement, directory, authority, CA
- X.509 authentication and certificates
- public key infrastructure (PKIX)
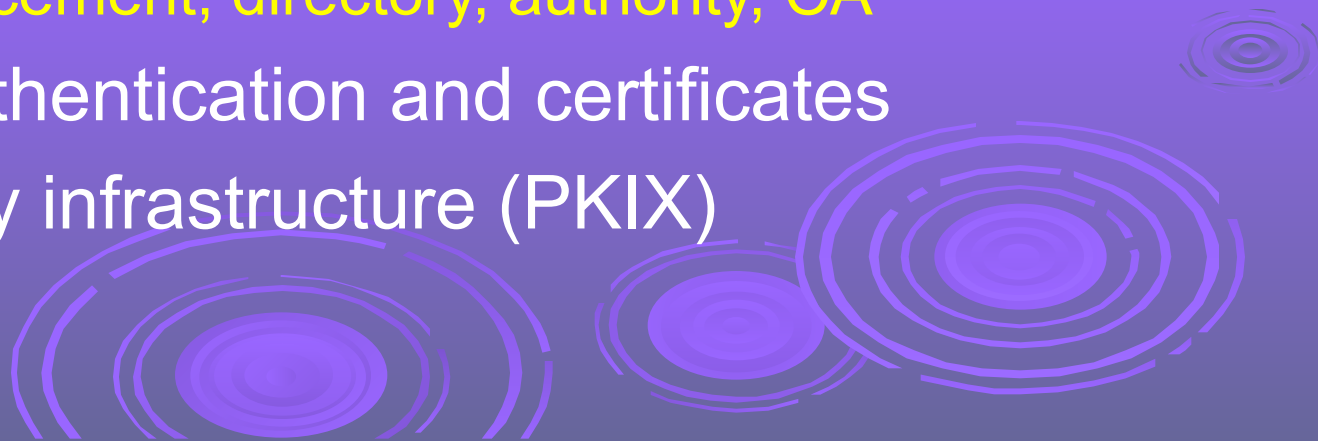
Distribution of Public Keys

⯐ can be considered as using one of:

- public announcement
- publicly available directory
- public-key authority
- public-key certificates
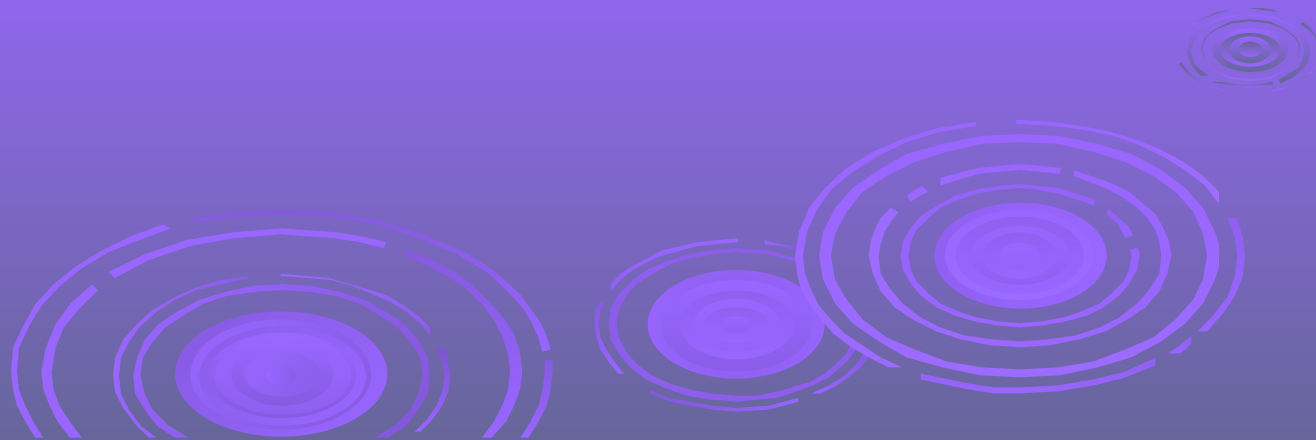
Public Announcement

- 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
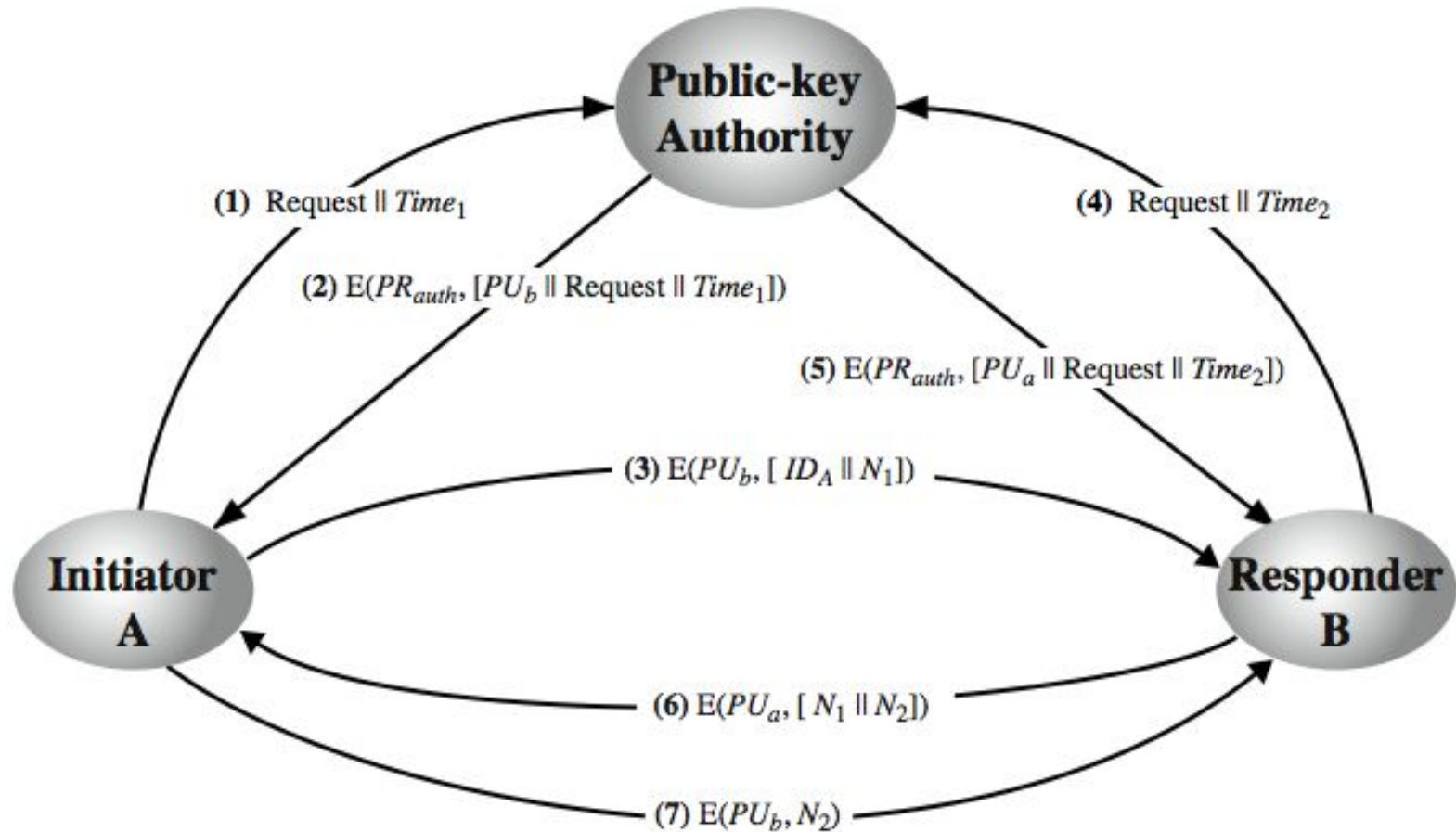 - until forgery is discovered can masquerade as claimed user

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

Public-Key Authority

Public-key Authority

(1) Request ∥ $Time_1$

(2) $E(PR_{auth}, [PU_b ∥ Request ∥ Time_1])$

(4) Request ∥ $Time_2$

(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) $E(PU_b, N_2)$

Road Map

- 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)

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
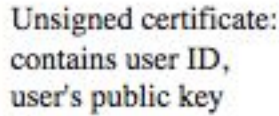
Public-Key Certificates
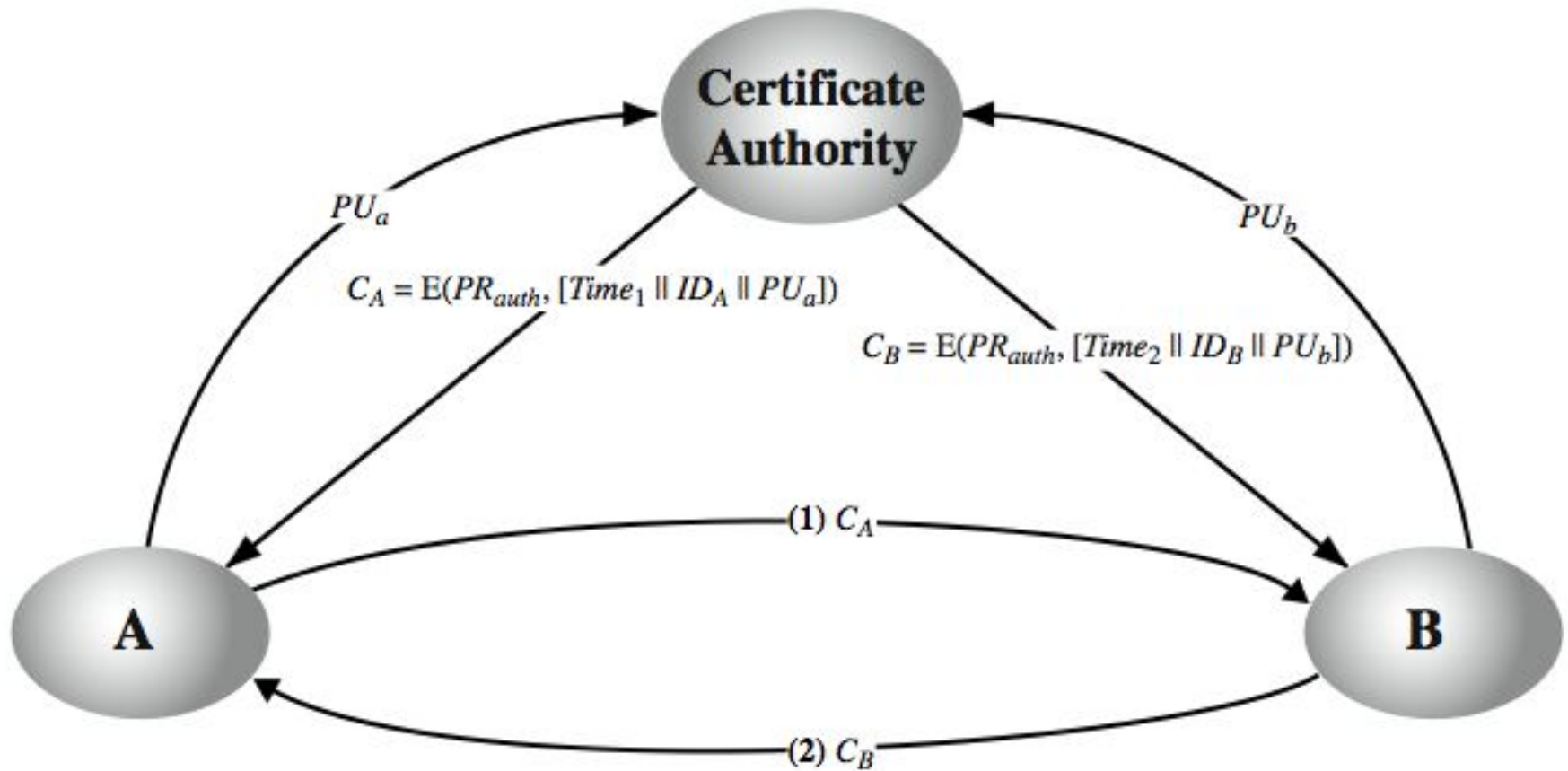
X.509 Authentication Service

- part of CCITT X.500 directory 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

Unsigned certificate: contains user ID, user's public key

Generate hash code of unsigned certificate

Sign hash code with CA's private key to form signature

Signed certificate: Recipient can verify signature using CA's public key.

X.509 Certificates

- issued by a Certification Authority (CA), containing:
    - version V (1, 2, or 3)
    - serial number SN (unique within CA) identifying certificate
    - 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)
    - issuer unique identifier (v2+)
    - subject unique identifier (v2+)
    - extension fields (v3)
    - signature (of hash of all fields in certificate)
- notation `CA<<A>>` denotes certificate for A signed by CA
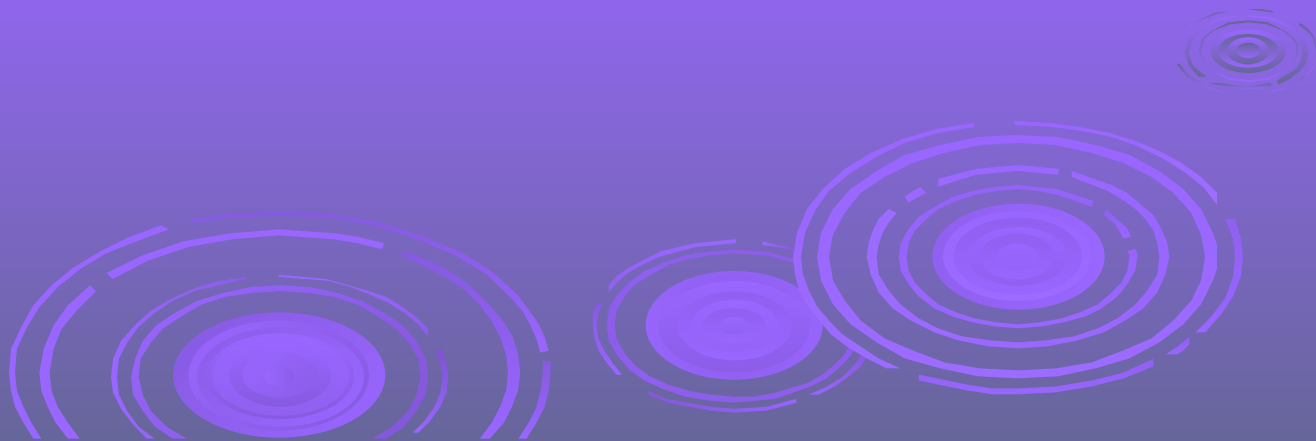
X.509 Certificates
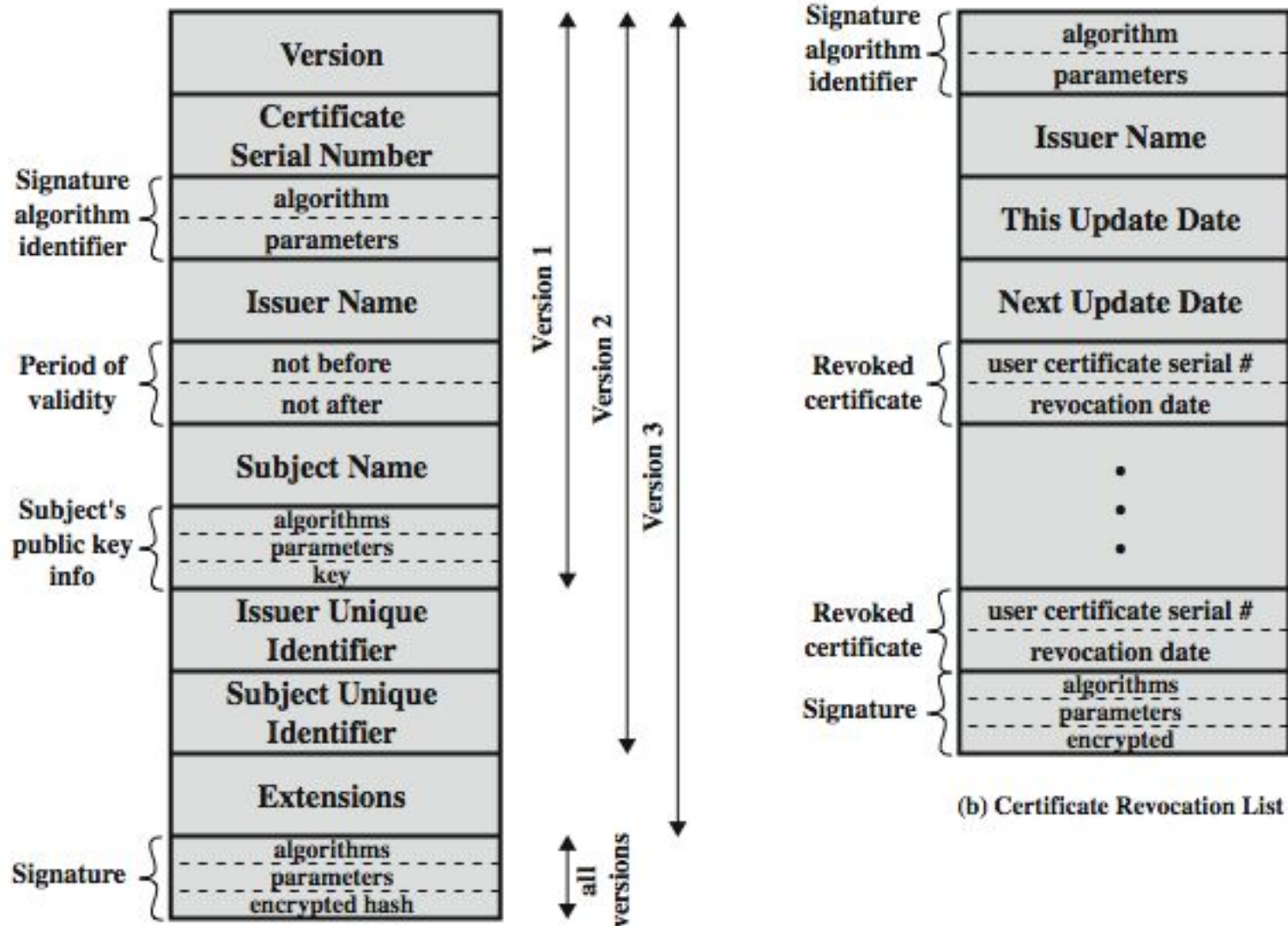
(a) X.509 Certificate

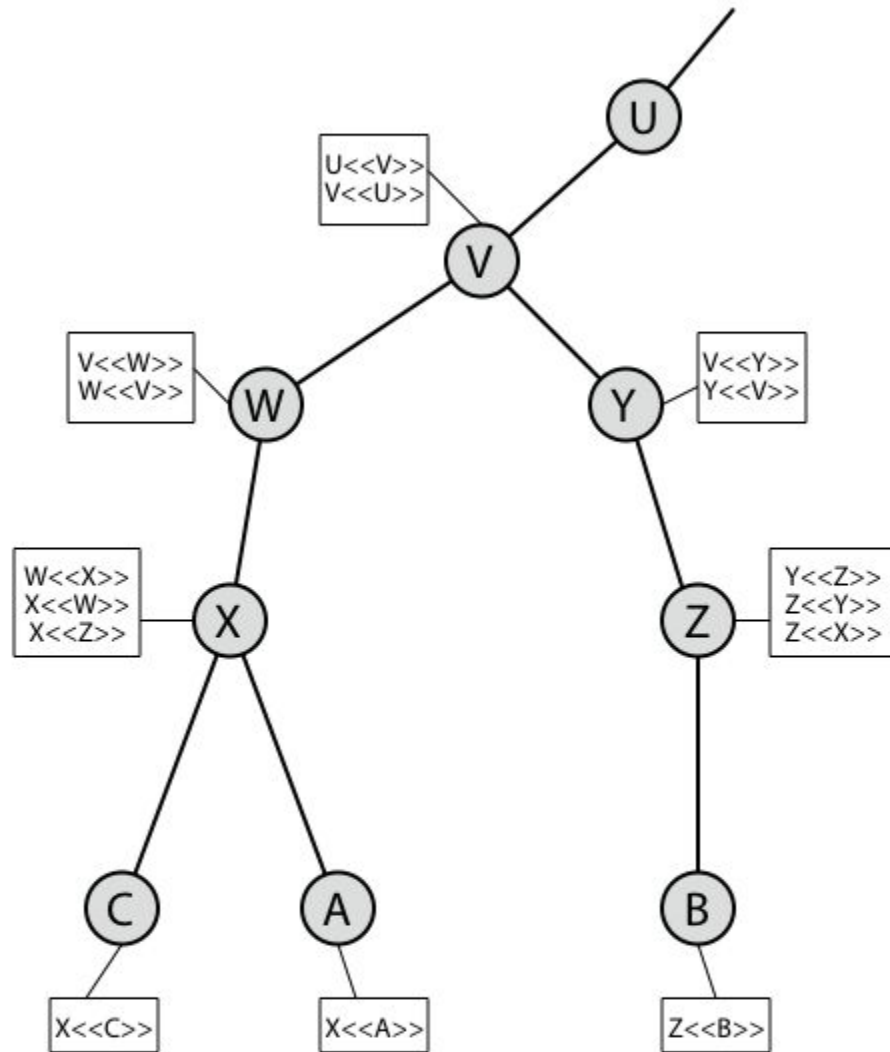(b) Certificate Revocation List

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

CA Hierarchy

- 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

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
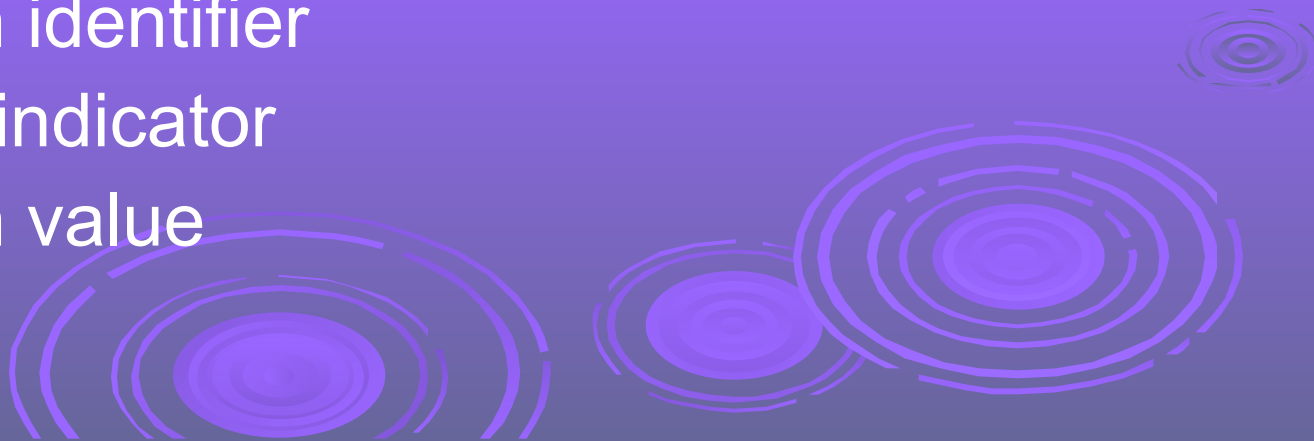
Certificate Extensions

- 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
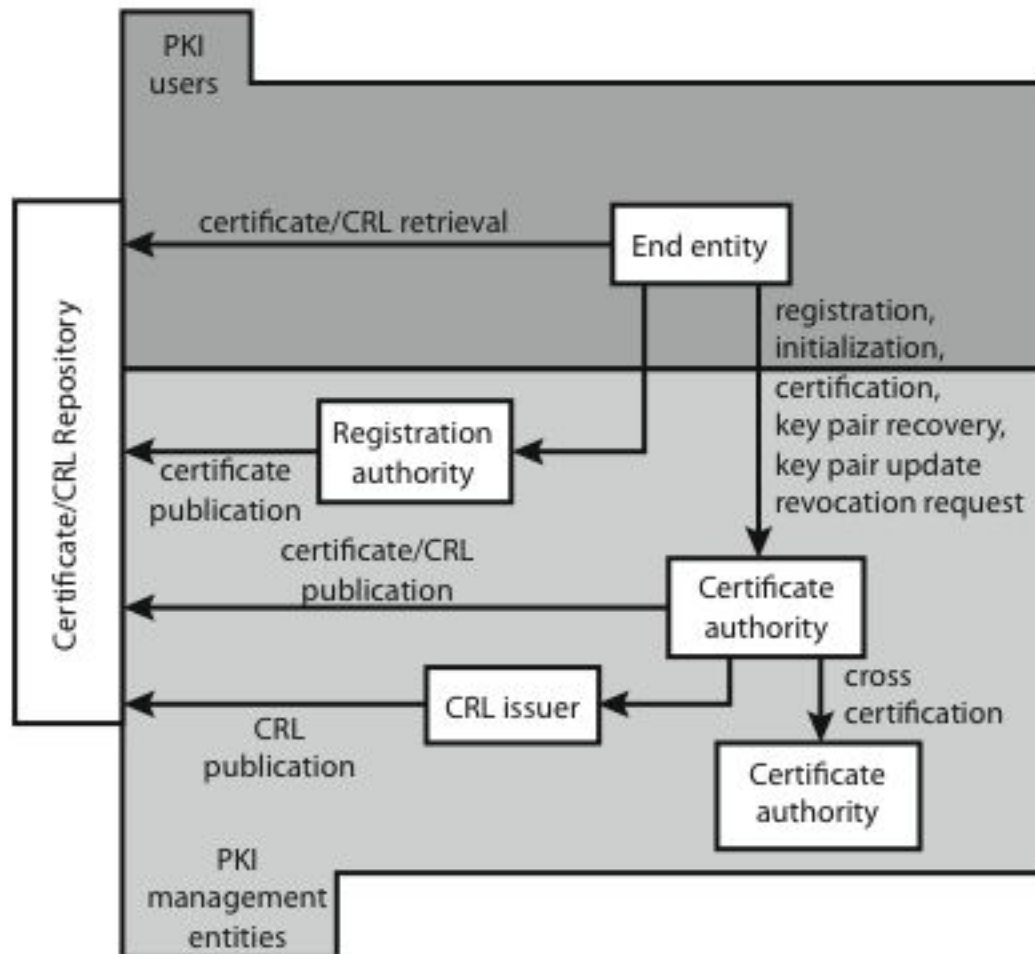 - allow constraints on use of certificates by other CA's
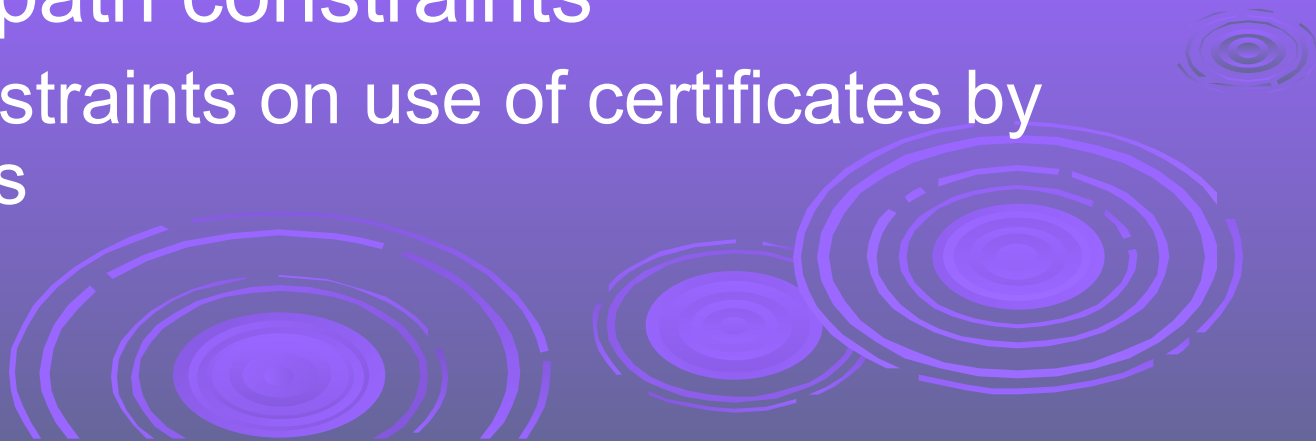
Road Map

- 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)
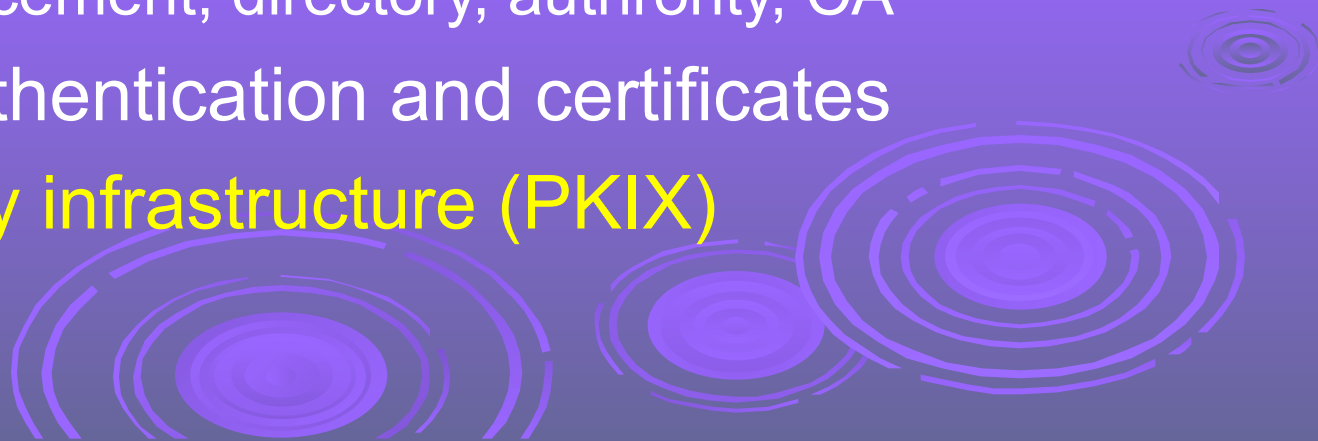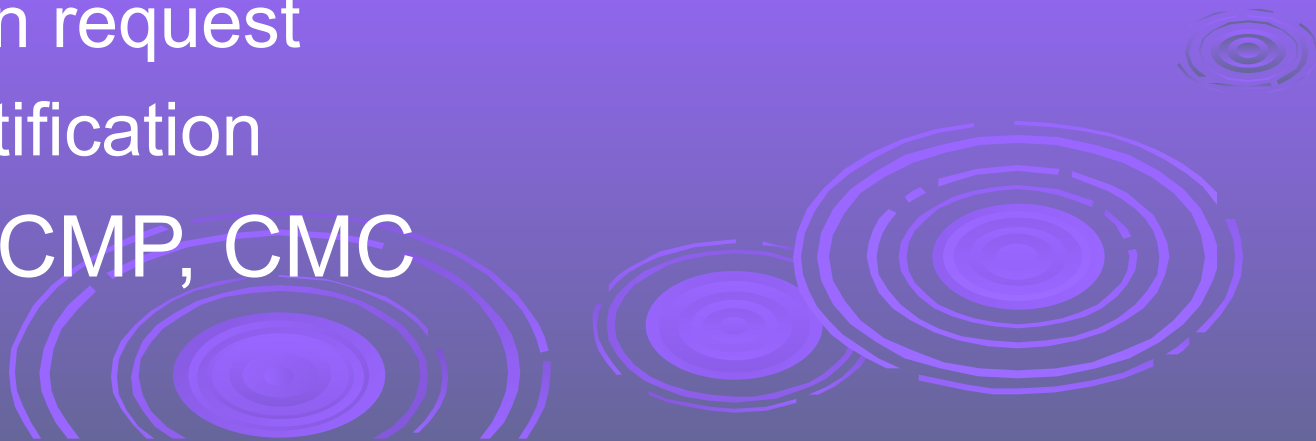
Public Key Infrastructure

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)