Key Management and Distribution

CSS322: Security and Cryptography

Sirindhorn International Institute of Technology
Thammasat University

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Contents

Key Distribution and Management

Symmetric Key Distribution using Symmetric Encryption

Symmetric Key Distribution using Asymmetric Encryption

Distribution of Public Keys

X.509 Certificates

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Key Distribution and Management

- ► Symmetric key cryptography: fast implementations, good for encrypting large amounts of data; requires shared secret key

- ► Asymmetric (public) key cryptography: inefficient for large data, good for authentication; no need to share a secret

- ► How to share symmetric keys?

- ► How to distribute public keys?

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Contents

Key Distribution and Management

Symmetric Key Distribution using Symmetric Encryption

Symmetric Key Distribution using Asymmetric Encryption

Distribution of Public Keys

X.509 Certificates

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Symmetric Key Distribution using Symmetric Encryption

- ▶ Objective: two entities share same secret key
- ▶ Principle: change keys frequently
- ▶ How to exchange a secret key?
    1. A physically delivers key to B
    2. Third party, C, can physically deliver key to A and B
    3. If A and B already have a key, can securely transmit new key to each other, encrypted with old key
    4. If A and B have secure connection with third party C, C can securely send keys to A and B
- ▶ Option 1 and 2: manual delivery; feasible if number of entites is small (link encryption)
- ▶ Option 3: requires initial distribution of key; discovery of initial key releases all subsequent keys
- ▶ Option 4: requires initial distribution of key with C; practical for large-scale systems (end-to-end encryption)

CSS322

Key Management
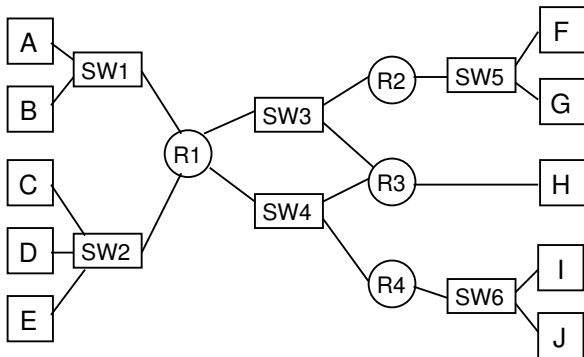
Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Link Encryption vs End-to-End Encryption

Link Encryption

- ▶ Encrypt data over individual links in network
- ▶ Each link end-point shares a secret key
- ▶ Decrypt/Encrypt at each device in path
- ▶ Requires all links/devices to support encryption

End-to-End Encryption

- ▶ Encrypt data at network end-points (e.g. hosts or applications)
- ▶ Each pair of hosts/applications share a secret key
- ▶ Does not rely on intermediate network devices

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

How Many Keys Need To Be Exchanged?

- ▶ Link-level encryption?
- ▶ End-to-end encryption between hosts?
- ▶ End-to-end encryption between applications?

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Using a Key Distribution Centre

- Key Distribution Centre (KDC) is trusted third party
- Hierarchy of keys used:
  - Data sent between end-systems encrypted with temporary session key
  - Session keys obtained from KDC over network; encrypted with master key
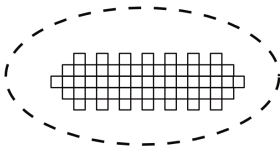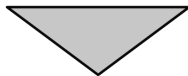 - Master keys can be distributed using manual delivery

CSS322

Key Management

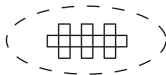Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Use of a Key Hierarchy

Data — Cryptographic Protection

Session Keys — Cryptographic Protection

Master Keys — Non-Cryptographic Protection

CSS322

Key Management

Key Distribution

Symmetric with Symmetric

Symmetric with Asymmetric

Public Keys

X.509

Key Distribution Scenario

Key Distribution Center (KDC)

(1) $ID_A \parallel ID_B \parallel N_1$

Key distribution steps

(2) $E(K_a, [K_s \parallel ID_A \parallel ID_B \parallel N_1]) \parallel E(K_b, [K_s \parallel ID_A])$

(3) $E(K_b, [K_s \parallel ID_A])$

Initiator A

Responder B

(4) $E(K_s, N_2)$

Authentication steps
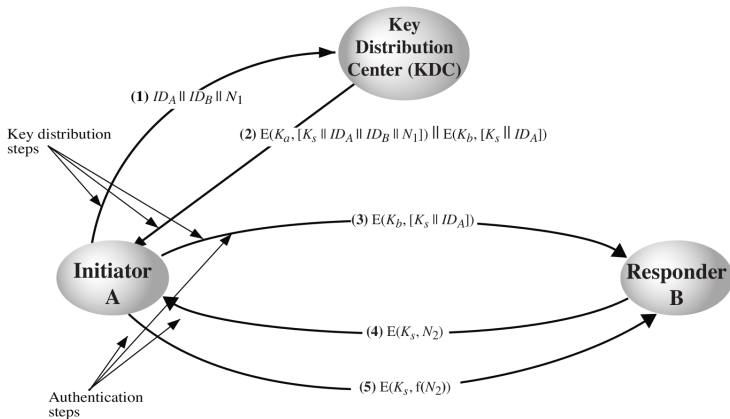
(5) $E(K_s, f(N_2))$

CSS322

Key Management

Key Distribution
Symmetric with
Symmetric
Symmetric with
Asymmetric
Public Keys
X.509

KDC Scenario Notation

- ▶ End-systems: A and B, identified by ID_A and ID_B
- ▶ Master keys: K_a, K_b
- ▶ Session key (between A and B): K_s
- ▶ Nonce values: N_1, N_2
    - ▶ E.g. timestamp, counter, random value
    - ▶ Must be different for each request
    - ▶ Must be difficult for attacker to guess

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Practical Considerations

Hierarchical Key Control

▶ Use multiple KDCs in a hierarchy

▶ E.g. KDC for each LAN (or building); central KDC to exchange keys between hosts in different LANs

▶ Reduces effort in key distribution; limits damage if local KDC is compromised

Session Key Lifetime

▶ Shorter lifetime is more secure; but increases overhead of exchanges

▶ Connection-oriented protocols (e.g. TCP): new session key for each connection

▶ Connection-less protocols (e.g. UDP/IP): change after fixed period or certain number of packets sent
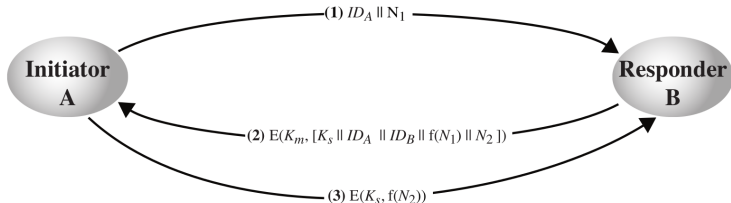
CSS322

Key Management

Key Distribution
Symmetric with
Symmetric
Symmetric with
Asymmetric
Public Keys
X.509

Decentralised Key Distribution

- Alternative that doesn't rely on KDC
- Each end-system must manually exchange $n - 1$ master keys (K_m) with others

(1) $ID_A \parallel N_1$

(2) $E(K_m, [K_s \parallel ID_A \parallel ID_B \parallel f(N_1) \parallel N_2])$

(3) $E(K_s, f(N_2))$

Initiator A

Responder B

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Contents

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys
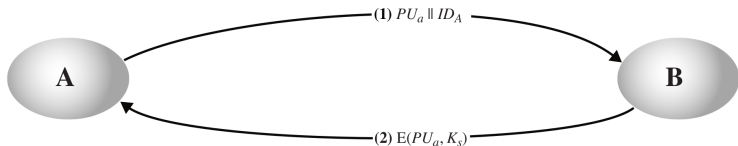
X.509

Symmetric Key Distribution using Asymmetric Encryption

- ▶ Asymmetric encryption generally too slow for encrypting large amount of data
- ▶ Common application of asymmetric encryption is exchanging secret keys
- ▶ Three examples:
    1. Simple Secret Key Distribution
    2. Secret Key Distribution with Confidentiality and Authentication
    3. Hybrid Scheme: Public-Key Distribution of KDC Master Keys

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

**Symmetric with
Asymmetric**

Public Keys

X.509

Simple Secret Key Distribution

- ▶ Simple: no keys prior to or after communication
- ▶ Provides confidentiality for session key
- ▶ Subject to man-in-the-middle attack
- ▶ Only useful if attacker cannot modify/insert messages

Man-in-the-Middle Attack

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Secret Key Distribution with Confidentiality and Authentication

▶ Provides both confidentiality and authentication in exchange of secret key

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

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

**Initiator
A**

**Responder
B**

(3) $E(PU_b, N_2)$

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

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509
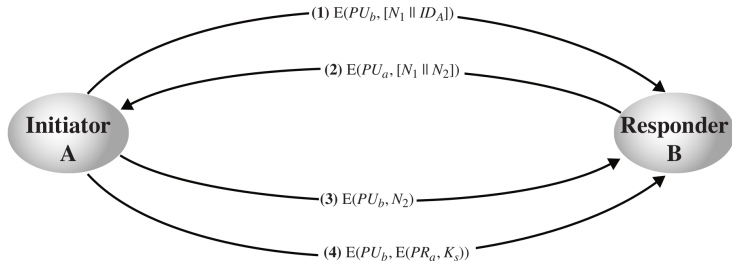
Hybrid Scheme: Public-Key Distribution of KDC Master Keys

- ▶ Use public-key distribution of secret keys when exchanging master keys between end-systems and KDC

- ▶ Efficient method of delivering master keys (rather than manual delivery)

- ▶ Useful for large networks, widely distributed set of users with single KDC

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Contents

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Distribution of Public Keys

▶ By design, public keys are made public

▶ Issue: how to ensure public key of A actually belongs to A (and not someone pretending to be A)

▶ Four approaches for distributing public keys
  1. Public announcement
  2. Publicly available directory
  3. Public-key authority
  4. Public-key certificates

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
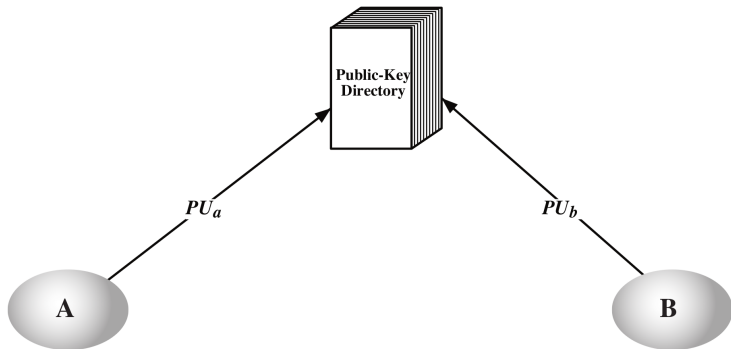Asymmetric

Public Keys

X.509

Public Announcements

- ▶ Make public key available in open forum: newspaper, email signature, website, conference, . . .
- ▶ Problem: anyone can announce a key pretending to be another user

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509
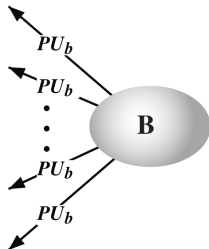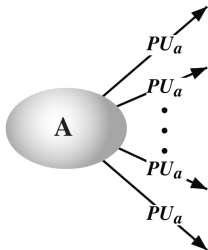
Publicly Available Directory

- All users publish keys in central directory
- Users must provide identification when publishing key
- Users can access directory electronically
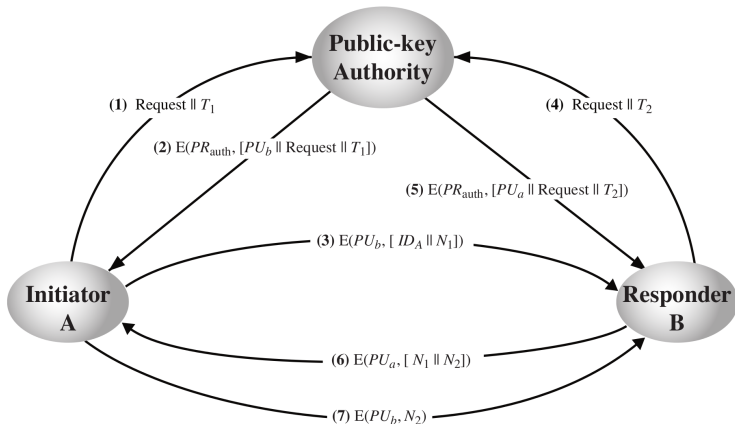- Weakness: directory must be secure

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Public-Key Authority

- Specific instance of using publicly available directory
- Assume each user has already security published public-key at authority; each user knows authorities public key

(1) Request ∥ T_1

(2) E(PR_{auth}, [PU_b ∥ Request ∥ T_1])

Public-key Authority

(4) Request ∥ T_2

(5) E(PR_{auth}, [PU_a ∥ Request ∥ T_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)

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Public-Key Authority

- First 5 messages are for key exchange; last 2 are authentication of users

- Although 7 messages, public keys obtained from authority can be cached

- Problem: authority can be bottleneck

- Alternative: public-key certificates

CSS322

Key Management

Key Distribution
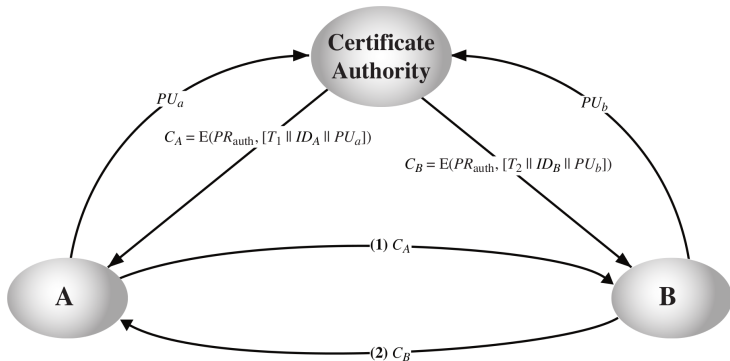
Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Public-Key Certificates

- Assume public keys sent to CA can be authenticated by CA; each user has certificate of CA

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Public Key Certificates

- A certificate is the ID and public-key of a user signed by CA

$$C_A = \mathrm{E}(PR_{auth}, [T||ID_A||PU_a])$$

- Timestamp T validates currency of certificate (expiration date)
- Common format for certificates is X.509 standard (by ITU)
  - S/MIME (secure email)
  - IP security (network layer security)
  - SSL/TLS (transport layer security)
  - SET (e-commerce)

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Contents

Key Distribution and Management

Symmetric Key Distribution using Symmetric Encryption

Symmetric Key Distribution using Asymmetric Encryption

Distribution of Public Keys

X.509 Certificates

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

X.509 Certificates

- ▶ Each user has a certificate, although it is created by the Certificate Authority (CA)
- ▶ Certificates are stored in a public directory
- ▶ Certificate format includes:
  - ▶ Version of X.509 certificate
  - ▶ Signature algorithm
  - ▶ CA's name and unique identifier
  - ▶ Period of validity
  - ▶ User's name and unique identifier
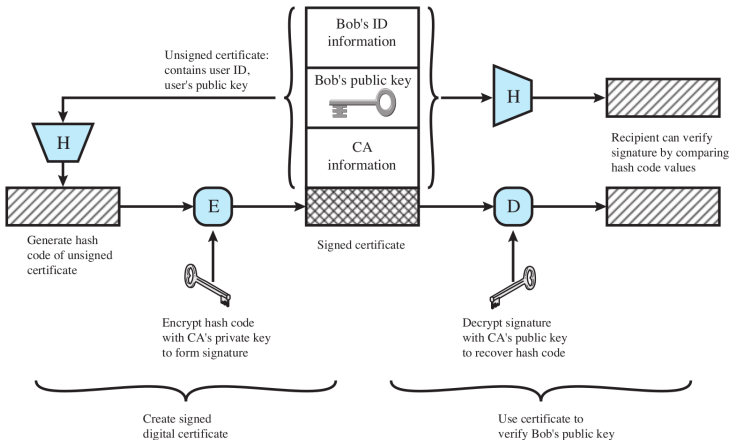 - ▶ User's public key information
  - ▶ Signature

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Public-Key Certificate Use

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
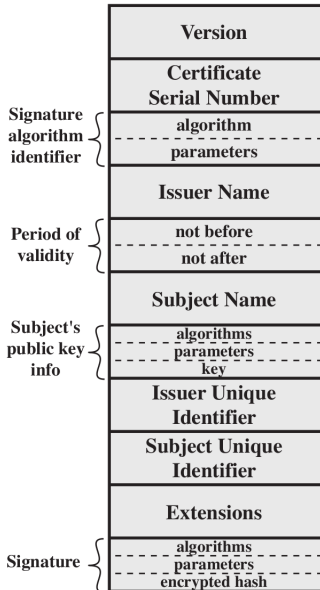Asymmetric

Public Keys

X.509

X.509 Formats

Version

Certificate
Serial Number

Signature
algorithm
identifier
- algorithm
- parameters

Issuer Name

Period of
validity
- not before
- not after

Subject Name

Subject's
public key
info
- algorithms
- parameters
- key

Issuer Unique
Identifier

Subject Unique
Identifier

Extensions

Signature
- algorithms
- parameters
- encrypted hash

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric
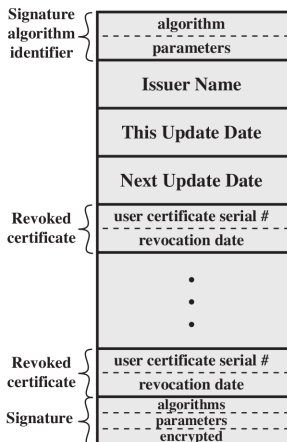
Public Keys

X.509

Certificate Revocation List

- ▶ Certificates may be revoked before expiry
- ▶ CA signs a CRL, which is stored in public directory

CSS322

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Multiple Certificate Authorities

- Multiple CA's can be arranged in hierarchy
- Notation: $Y << X >>$ certificate of X issued by CA Y