CSS441

Key Management

Key Distribution

Symmetric with Symmetric

Symmetric with Asymmetric

Public Keys

X.509

Key Management and Distribution

CSS441: Security and Cryptography

Sirindhorn International Institute of Technology
Thammasat University

1

CSS441

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

2

# Key Management

Challenges

- ▶ How to share a secret key?
- ▶ How to obtain someone else's public key?
- ▶ When to change keys?

Assumptions and Principles

- ▶ Many users wish to communicate securely across network
- ▶ Attacker can intercept any location in network
- ▶ Manual interactions between users are undesirable (e.g. physical exchange of keys)
- ▶ More times a key is used, greater chance for attacker to discover the key

Where Should Encryption Be Performed?
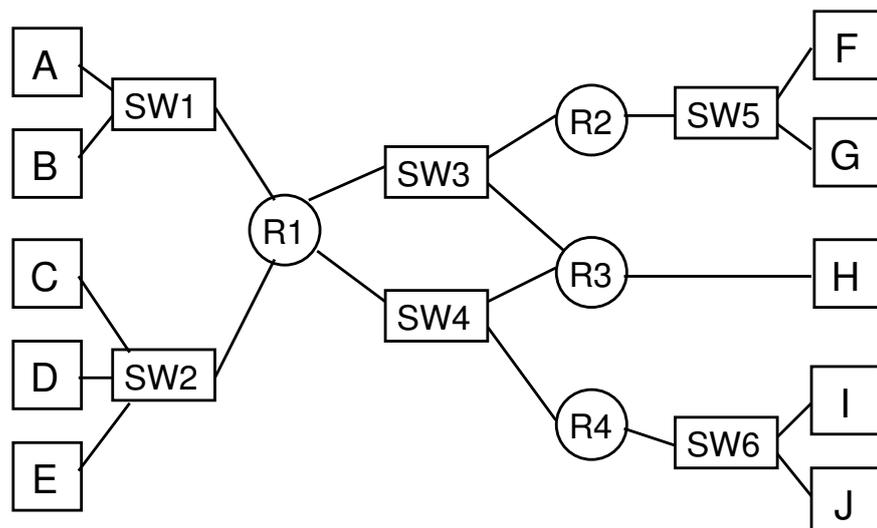
- ▶ Number of keys to be exchanged depends on number of entities wishing to communicate
- ▶ Related issue: where to perform encryption
  - ▶ Encrypt separately across each link
  - ▶ Encrypt only at end-points
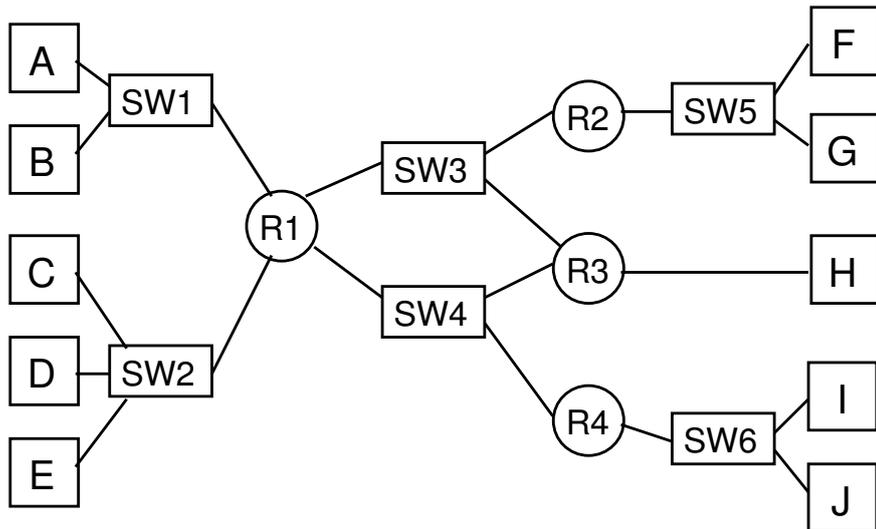
Link Encryption vs End-to-End Encryption

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

How Many Keys Need To Be Exchanged?

▶ Link-level encryption?

▶ End-to-end encryption between hosts?

▶ End-to-end encryption between applications?

CSS441

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Exchanging Secret Keys

Option 1: Manual Exchange of All Keys

- ▶ All users exchange secret keys with all other users manually (e.g. face-to-face)
- ▶ Inconvenient

Option 2: Manual Exchange of Master Keys

- ▶ All users exchange master key with trusted, central entity (e.g. Key Distribution Centre)
- ▶ Session keys automatically exchanged between users via KDC
- ▶ Security and performance bottleneck at KDC

CSS441

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Exchanging Secret Keys

Option 3: Public Key Cryptography to Exchange Secrets

- ▶ Use public-key cryptography to securely and automatically exchange secret keys
- ▶ Example 1: user A encrypts secret with user B's public key; sends to B
- ▶ Example 2: Diffie-Hellman secret key exchange
- ▶ Related issue: How to obtain someone else's public key?

Contents

Key Distribution and Management

Symmetric Key Distribution using Symmetric Encryption

Symmetric Key Distribution using Asymmetric Encryption

Distribution of Public Keys

X.509 Certificates

Symmetric Key Distribution using Symmetric Encryption

- ▶ Objective: two entities share same secret key
- ▶ Principle: change keys frequently
- ▶ How to exchange a secret key?
  1. Decentralised Key Distribution: manual distribution of master keys between all entities, automatic distribution of session keys
  2. Key Distribution Centre (KDC): manual distribution of master keys with KDC, automatic distribution of session keys

CSS441

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Key Hierarchy and Lifetimes

- ▶ Master keys used to securely exchange session keys

- ▶ Session keys used to securely exchange data

- ▶ Change session keys automatically and regularly

- ▶ Change master keys manually and seldom

- ▶ 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

CSS441

Key Management

Key Distribution

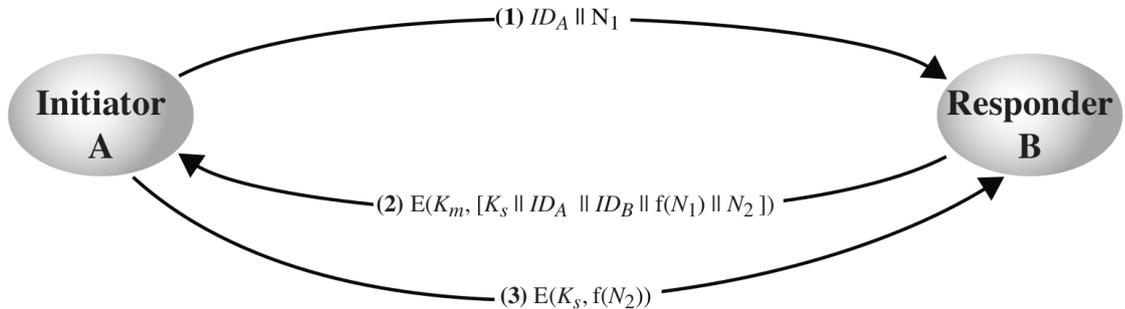Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Notation

- ▶ End-systems: A and B, identified by ID_A and ID_B

- ▶ Master key (between A and B): K_m

- ▶ Master keys specific to user: K_a, K_b

- ▶ Session key (between A and B): K_s

- ▶ Nonce values: N_1, N_2
  - ▶ Number used only once
  - ▶ E.g. time-stamp, counter, random value, function $f()$
  - ▶ Must be different for each request
  - ▶ Must be difficult for attacker to guess

CSS441

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Decentralised Key Distribution

- Each end-system must manually exchange $n - 1$ master keys (K_m) with others
- Does not rely on trusted-third party

(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

Credit: Figure 14.5 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

13

CSS441

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
- Users manually exchange master keys with KDC
- Users automatically obtain session key (via KDC) to communicate with other users
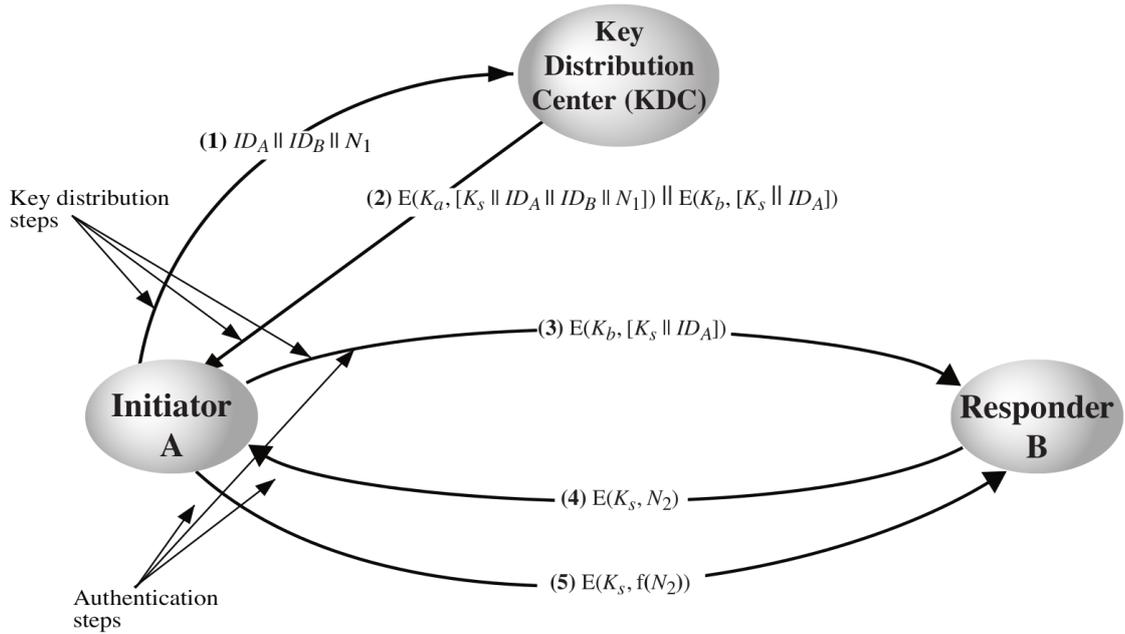
CSS441

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

Key Distribution with KDC

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

Key distribution
steps

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

Key Distribution Center (KDC)

Initiator A

Responder B

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

(4) $E(K_s, N_2)$

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

Authentication
steps

Credit: Figure 14.3 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

15

CSS441

Key Management

Key Distribution

Symmetric with
Symmetric

Symmetric with
Asymmetric

Public Keys

X.509

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

16

CSS441

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

CSS441

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

Simple Secret Key Distribution

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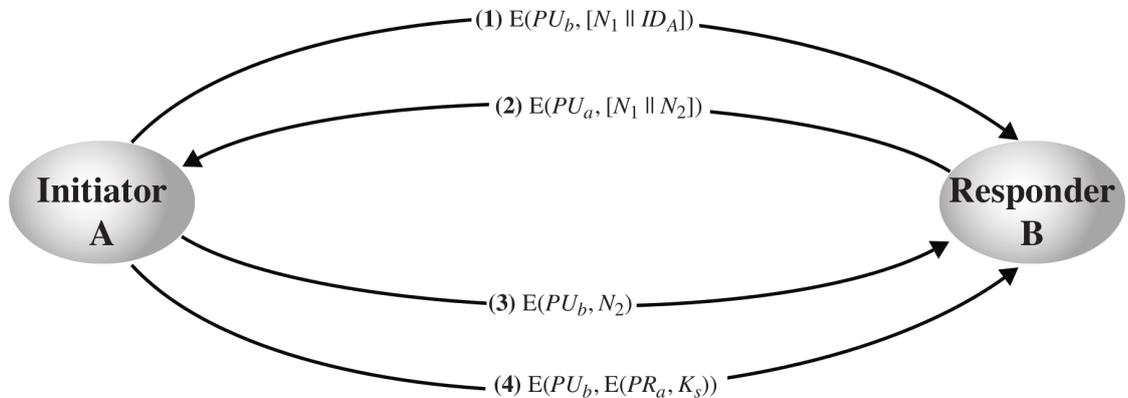- ► Only useful if attacker cannot modify/insert messages

$$\text{A} \quad \xrightarrow{\textbf{(1)} \ PU_a \parallel ID_A} \quad \text{B}$$
$$\text{A} \quad \xleftarrow{\textbf{(2)} \ E(PU_a, K_s)} \quad \text{B}$$
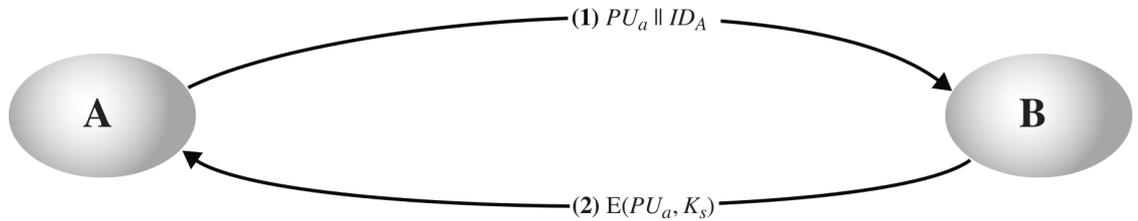
Credit: Figure 14.7 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

19

Man-in-the-Middle Attack

CSS441

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

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

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

Initiator A

Responder B

$$\text{(3) } E(PU_b, N_2)$$

$$\text{(4) } E(PU_b, E(PR_a, K_s))$$

Credit: Figure 14.8 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

21

CSS441

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

22

Contents

Key Distribution and Management

Symmetric Key Distribution using Symmetric Encryption

Symmetric Key Distribution using Asymmetric Encryption

Distribution of Public Keys

X.509 Certificates

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

CSS441

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

Credit: Figure 14.9 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

CSS441

Key Management

Key Distribution

Symmetric with
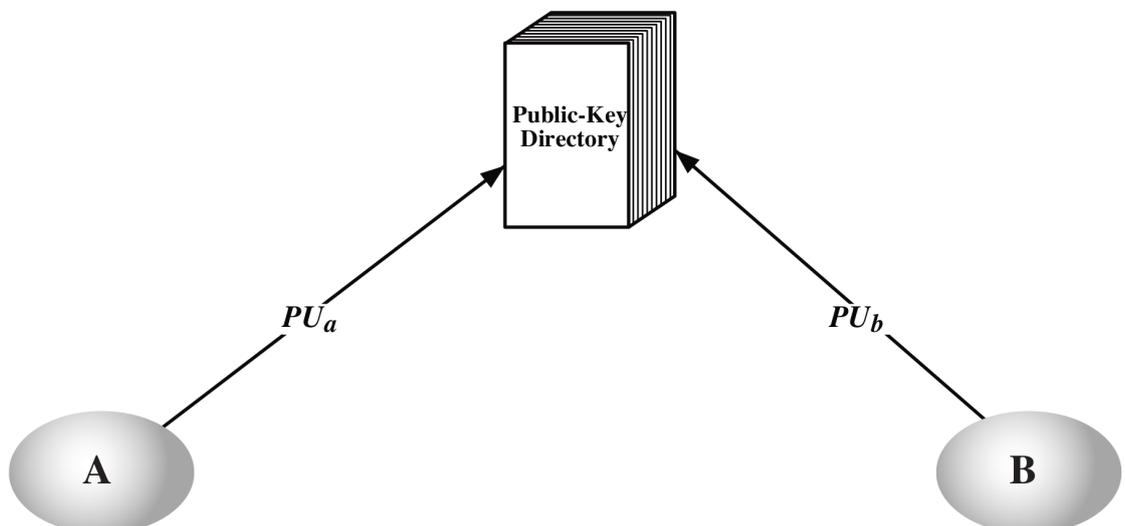Symmetric

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

Credit: Figure 14.10 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011
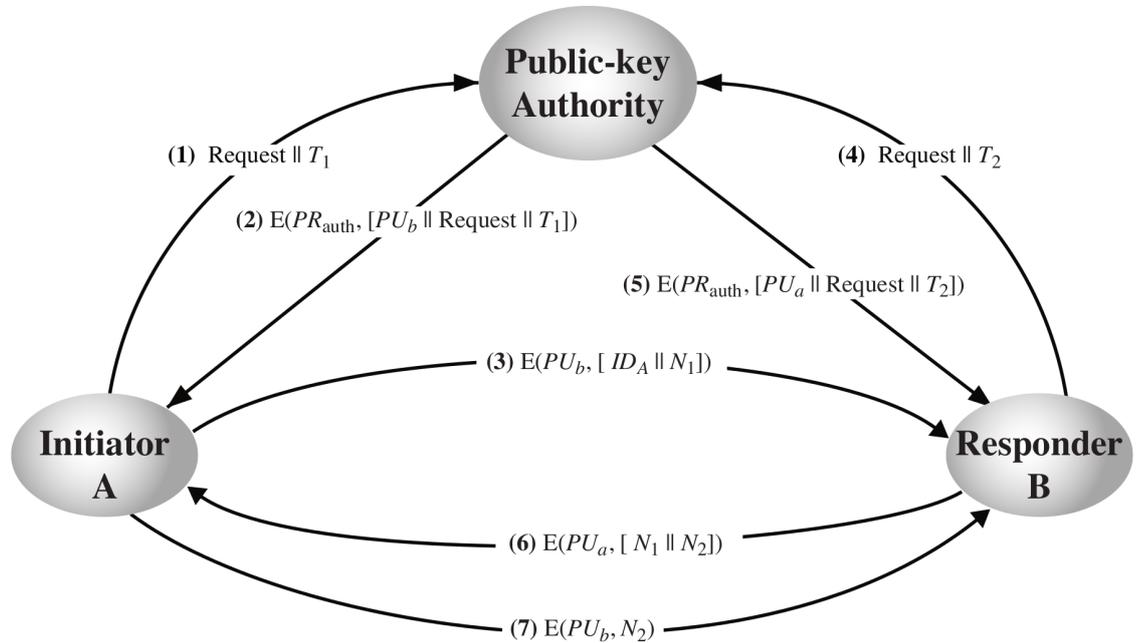
Public-Key Authority

- ▶ 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) $E(PR_{auth}, [PU_b ∥ Request ∥ T_1])$

(3) $E(PU_b, [ID_A ∥ N_1])$

(4) Request ∥ T_2

(5) $E(PR_{auth}, [PU_a ∥ Request ∥ T_2])$

(6) $E(PU_a, [N_1 ∥ N_2])$

(7) $E(PU_b, N_2)$

Public-key Authority

Initiator A

Responder B

Credit: Figure 14.11 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

27

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

CSS441

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
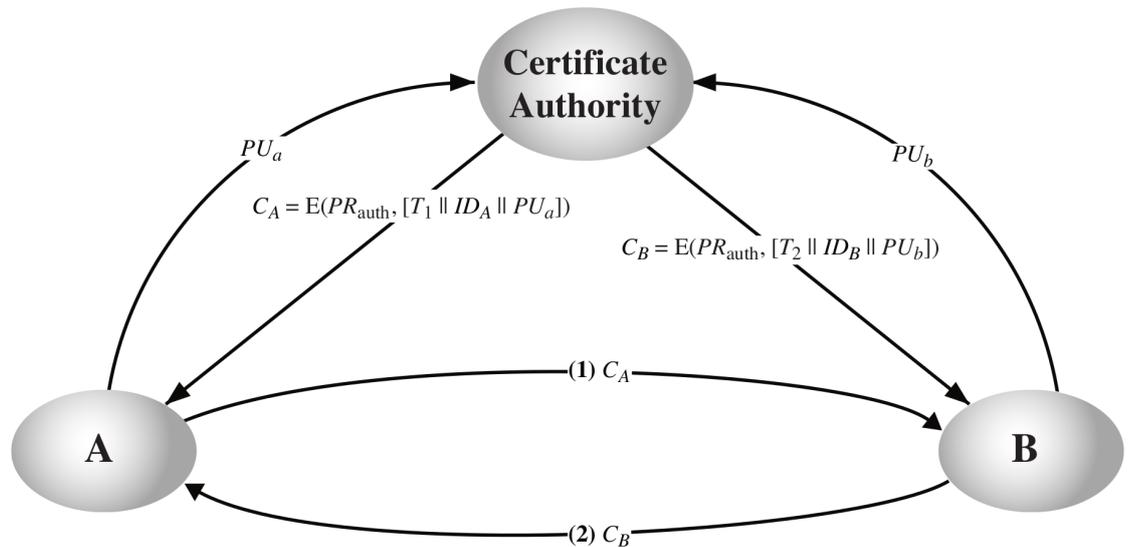
Credit: Figure 14.12 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

CSS441

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

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

▶ Time-stamp 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)

CSS441

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

CSS441

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
  - ▶ Serial number unique to the issuer (CA)
  - ▶ Signature algorithm
  - ▶ Issuer's name and unique identifier
  - ▶ Period of validity
  - ▶ Subject's name and unique identifier
  - ▶ Subject's public key information: algorithm, parameters, key
  - ▶ Signature

▶ Certificates may be revoked before expiry
  - ▶ CA signs a Certificate Revocation List (CRL), which is stored in public directory

CSS441

Key Management

Key Distribution

Symmetric with
Symmetric
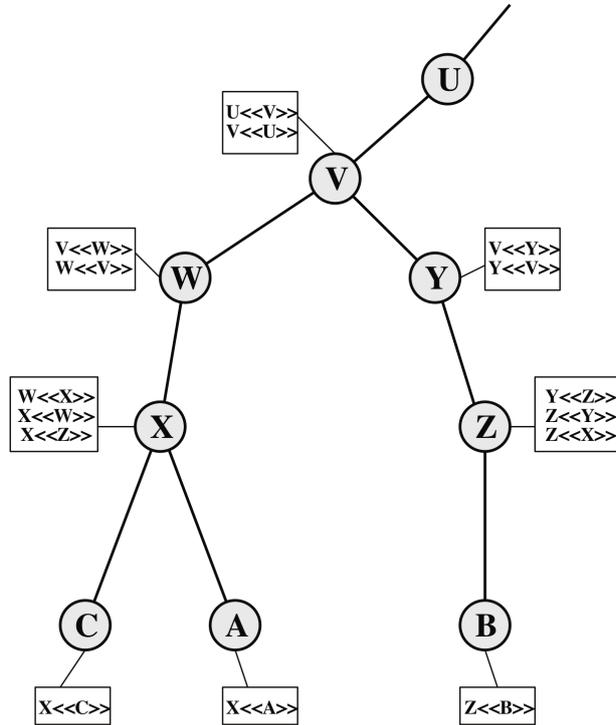
Symmetric with
Asymmetric

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

Credit: Figure 14.15 in Stallings, *Cryptography and Network Security*, 5th Ed., Pearson 2011

33