Network Security (NetSec)

Summer 2015
Chapter 03: Application Level Security
Module 04: Key Mgmt. & Certificates

Prof. Dr.-Ing. Matthias Hollick

Technische Universität Darmstadt
Secure Mobile Networking Lab - SEEMOO
Department of Computer Science
Center for Advanced Security Research Darmstadt - CASED

Mornewegstr. 32
D-64293 Darmstadt, Germany
Tel.+49 6151 16-70922, Fax. +49 6151 16-70921
http://seemoo.de or http://www.seemoo.tu-darmstadt.de

Prof. Dr.-Ing. Matthias Hollick
matthias.hollick@seemoo.tu-darmstadt.de

Learning Objectives & Outline

Learning objectives

- Application and transport level security need mechanisms to exchange keys. Here: understand the role of key management and distribution
- Give broad overview on this topic

Outline

(1) The need for scalable key distribution and management
(2) public announcement of keys
(3) publicly available directory of keys
(4) public-key authority
(5) public-key certificates (and certificate authorities)
(6) Public Key Infrastructures: Cure-all or Disease?

Chapter 03, Module 04

Motivation: Key Management and Distribution

Key management and distribution is often Achilles heel of systems that rely on crypto

- often secure system failure due to problems in the key distribution scheme

We have discovered the need to associate keys with individuals or servers in the previous lectures (e.g. in PGP, to come in SSL)

- symmetric schemes require both parties to share common secret key
- issue is how to securely distribute this key

Question: How many keys do you need to exchange in a system with N entities?

[N(N – 1)]/2

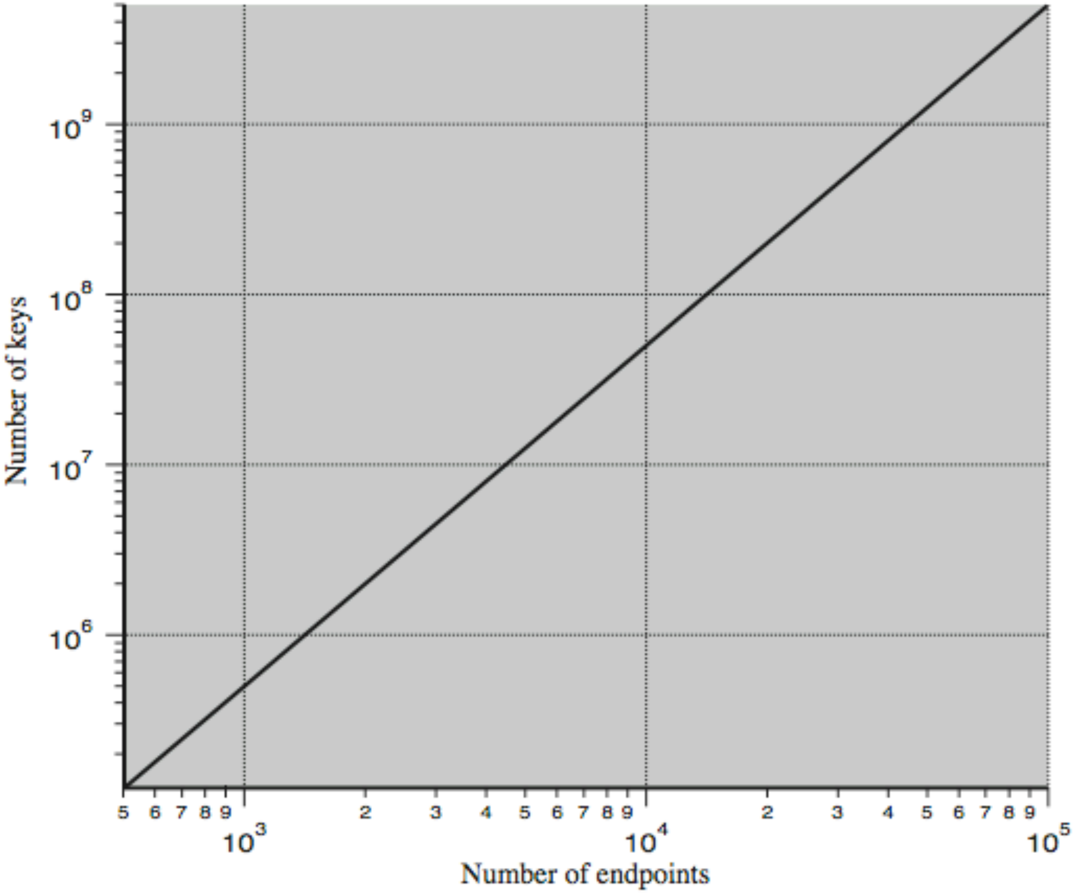
Key Distribution Task

Endpoints can be

- Hosts for network level security
- Users/processes for application level security

- For end-to-end encryption need to establish key for each pair of entitites that communicate

- I.e. ten thousand nodes require ~50,000,000 keys if all nodes want to communicate with each other



Key Distribution: Options

Given parties (A)lice and (B)ob have various key distribution alternatives:

Physical delivery

- A can select key and physically deliver to B
  - Scales poorly, personal contact necessary
- Third party can select & deliver key to A & B
  - Scales poorly, personal contact necessary
- If A & B have communicated previously can use previous key to encrypt a new key
  - Based on the above, can be breached, if attacker gets hold of secret

- If A & B have secure communications with a third party C, C can relay key between A & B

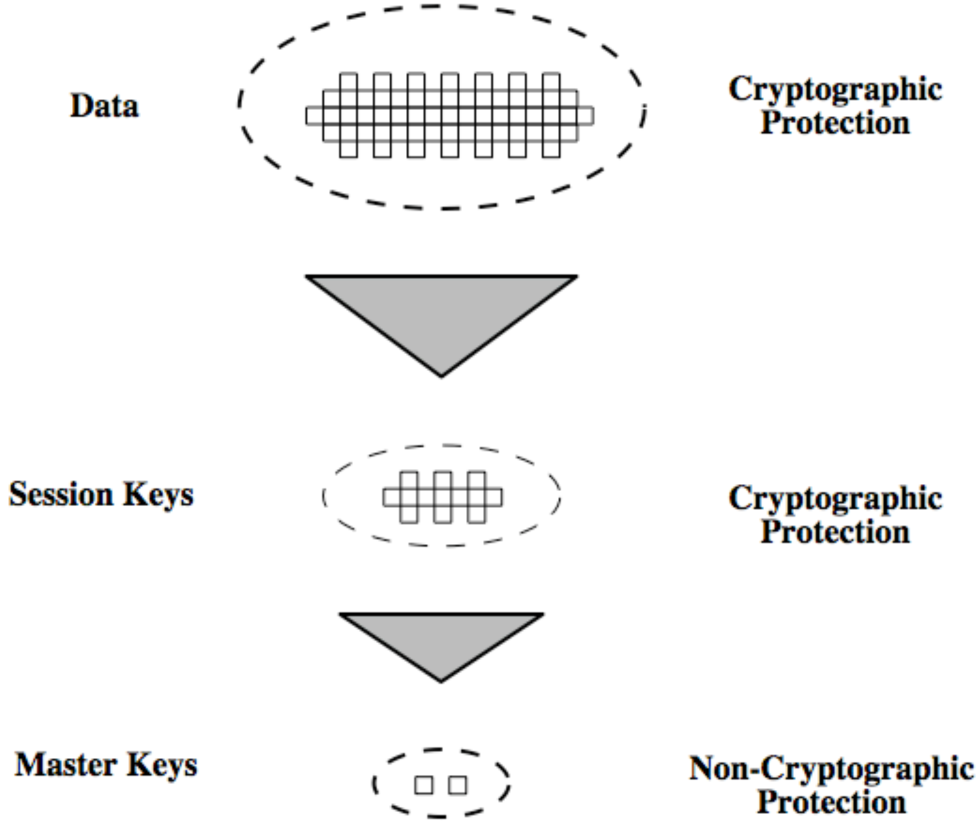
Key Hierarchy

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



Distribution of Public Keys

The following principles exist:

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

Image source: www.sxc.hu

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

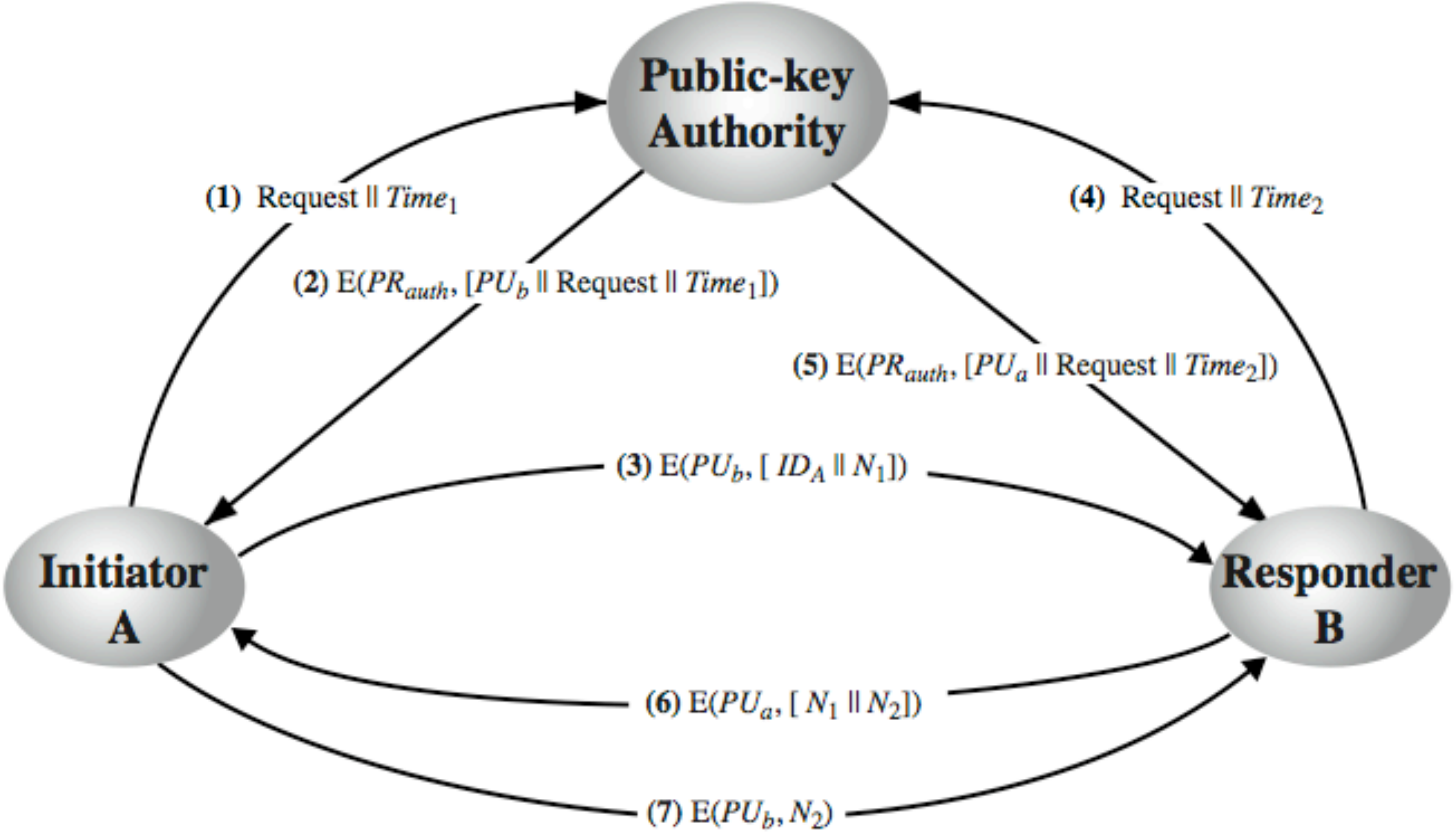
Public-Key Authority

Idea: 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)

Public-Key Certificates

Certificates allow key exchange without real-time access to public-key authority

a certificate binds identity to public key

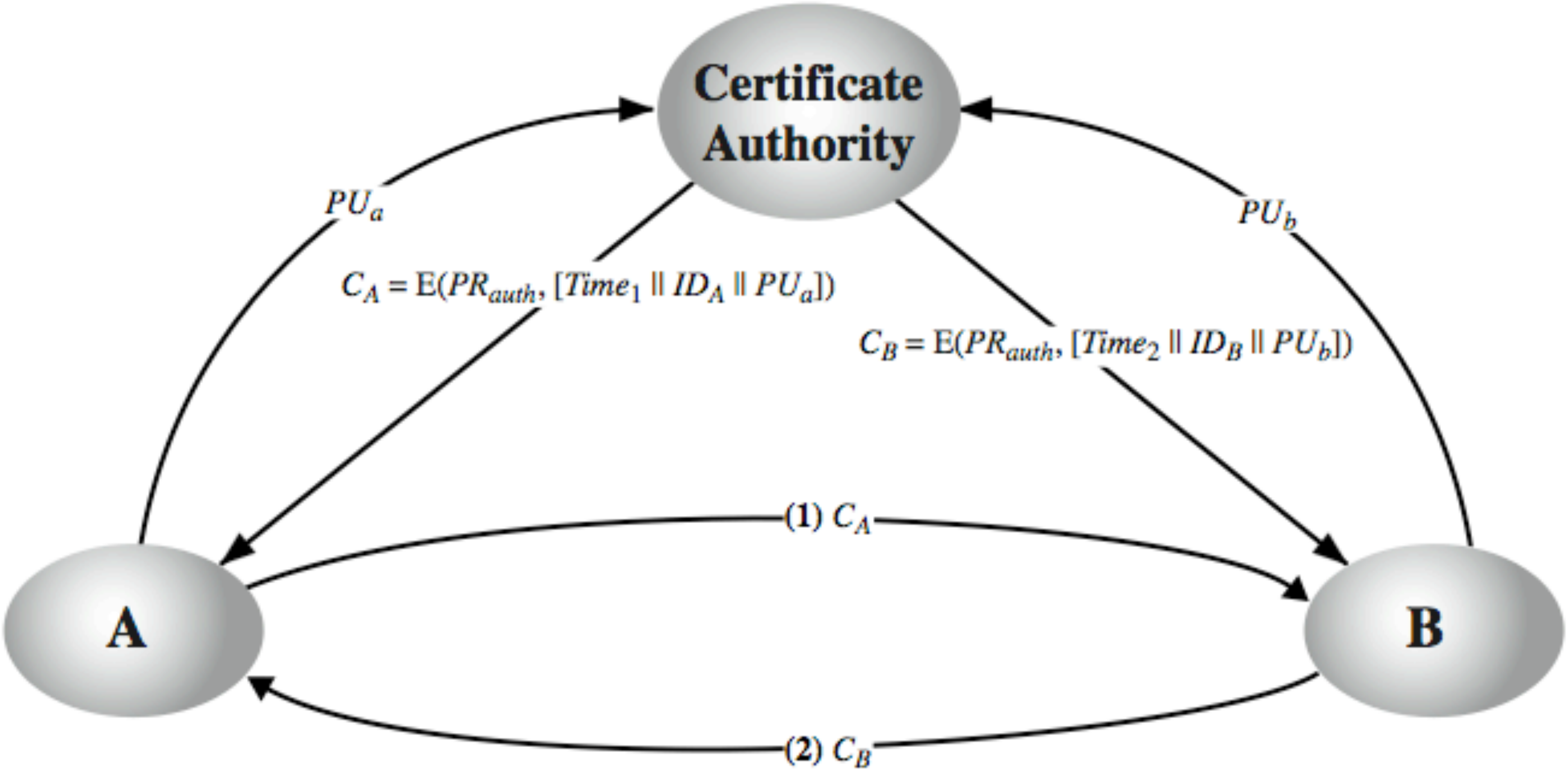
- usually with other info such as period of validity, rights of use etc

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

Universally adopted with the X.509 standard

Public-Key Certificates


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

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

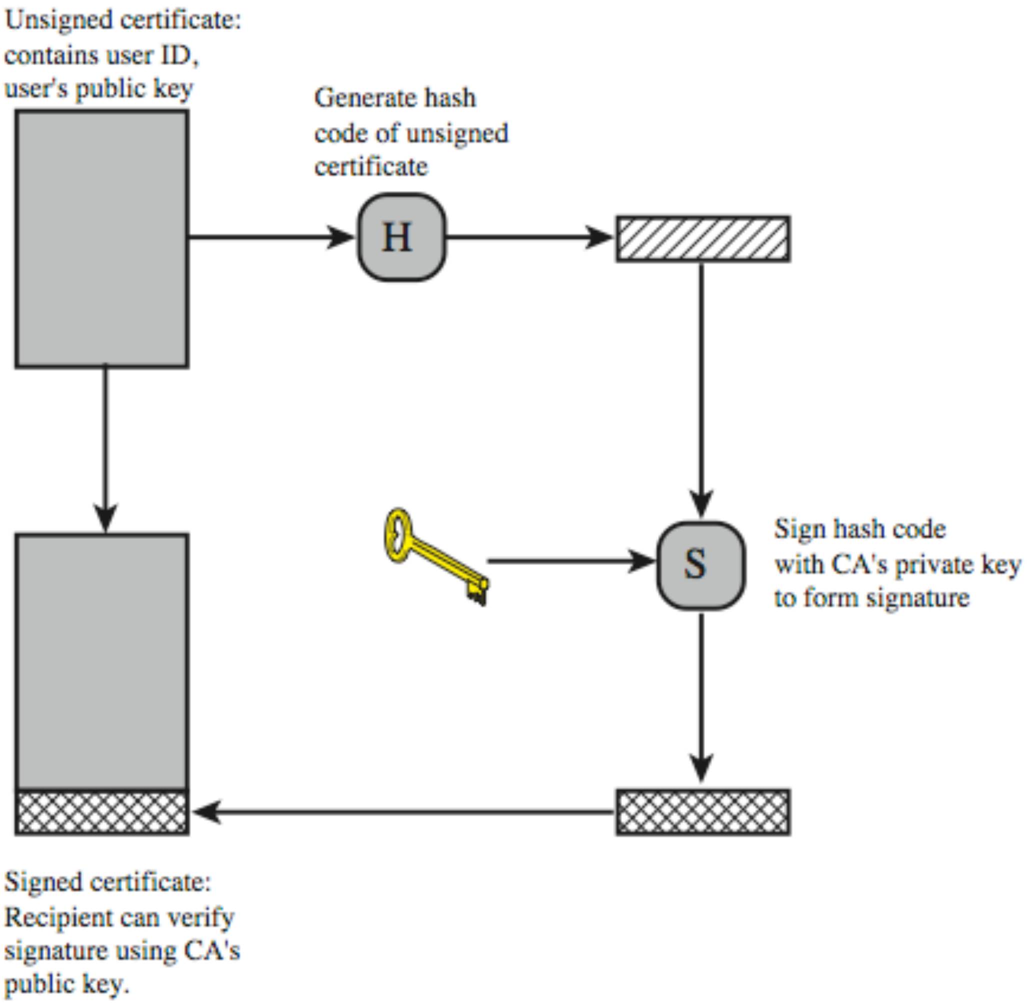
uses public-key crypto & digital signatures

- algorithms not standardised, but RSA recommended

X.509 certificates are widely used

- have 3 versions

X.509 Certificate Use



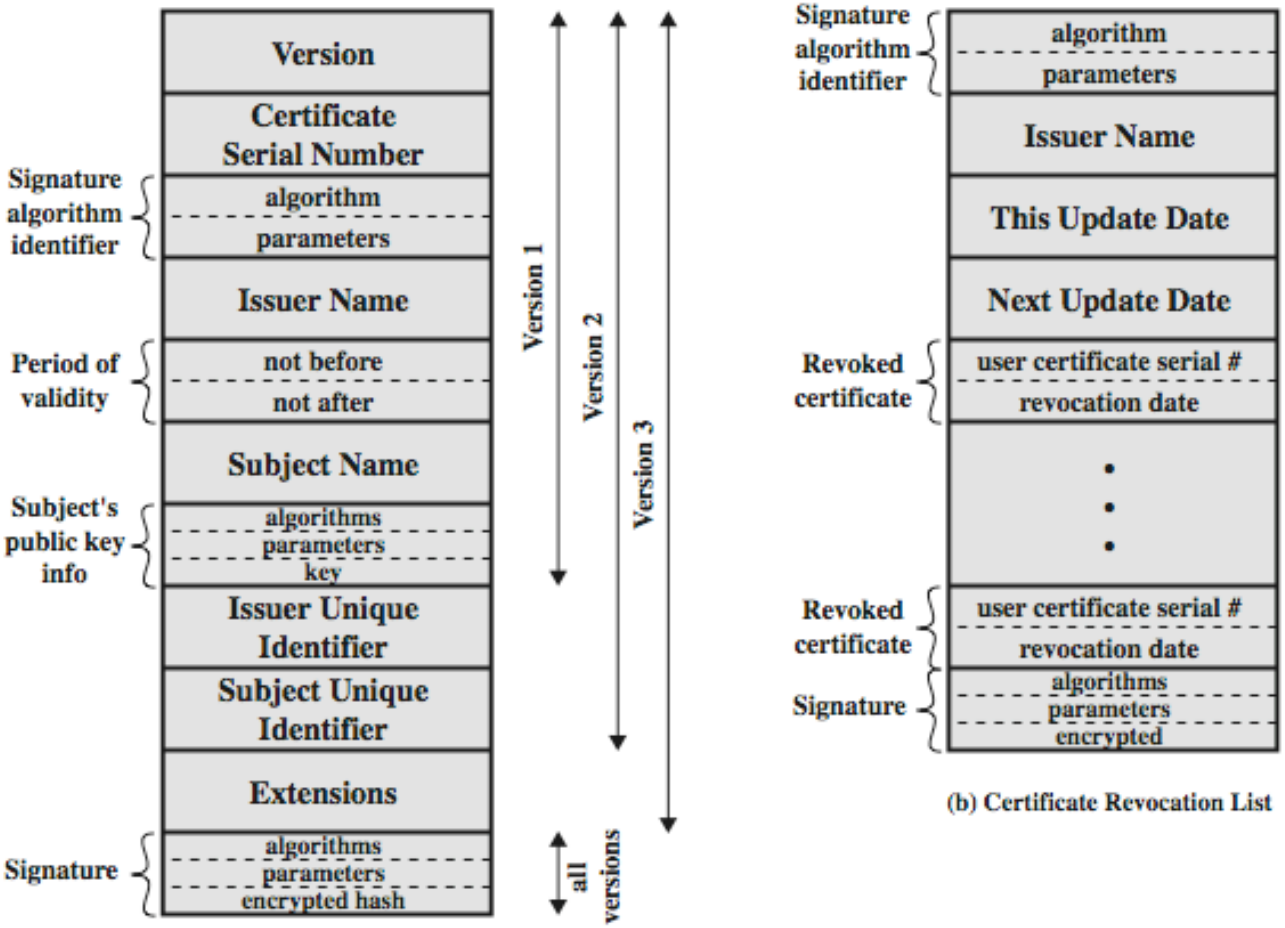
X.509 Certificates

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



(b) Certificate Revocation List

(a) X.509 Certificate

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

Certificate Revocation

certificates have a period of validity

may need to revoke before expiry, eg:

- user's private key is compromised
- user is no longer certified by this CA
- 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

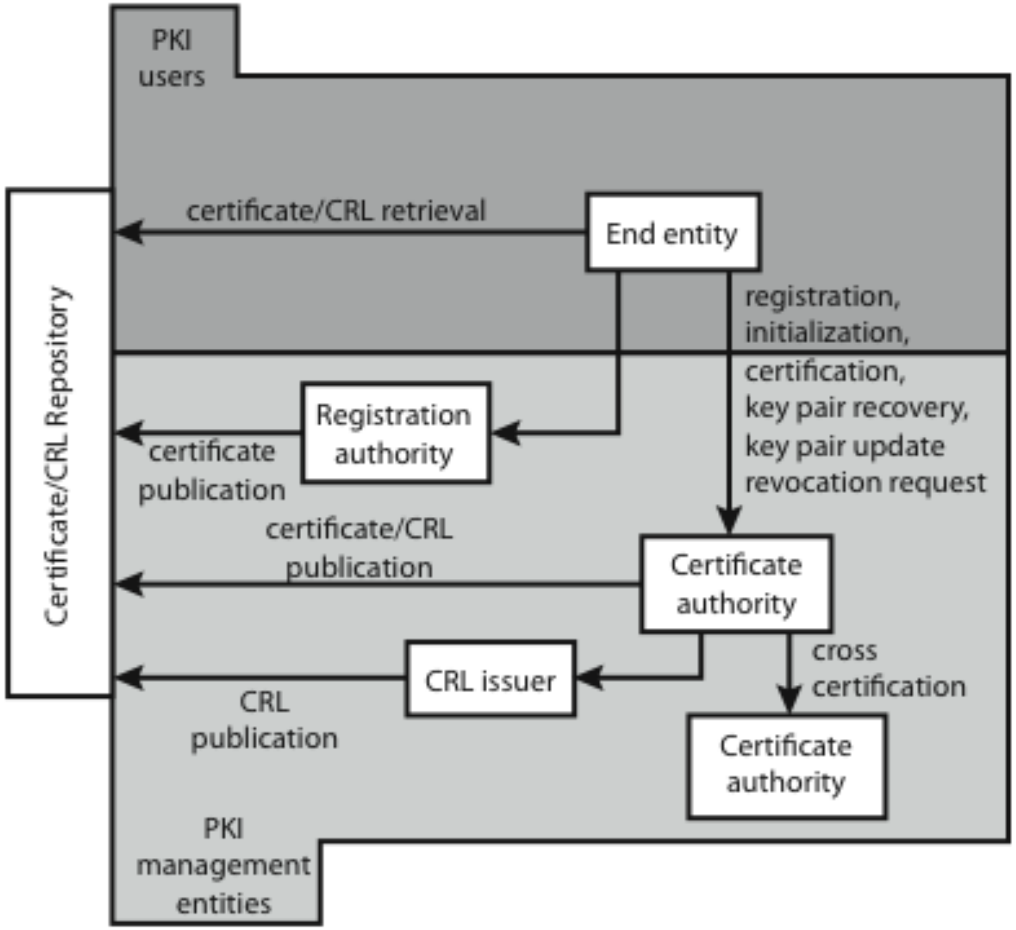
Public Key Infrastructure (PKIX)

PKIX

functions:

- registration
- initialization
- certification
- key pair recovery
- key pair update
- revocation request
- cross certification

protocols: CMP, CMC



PKI users
certificate/CRL retrieval
End entity
registration, initialization, certification, key pair recovery, key pair update revocation request
Certificate/CRL Repository
Registration authority
certificate publication
certificate/CRL publication
Certificate authority
CRL issuer
cross certification
CRL publication
Certificate authority
PKI management entities

Public Key Infrastructures: Cure-all or Disease

Which problems do you know w.r.t. PKIs?

Summary

Have considered:

- distribution of public keys
  - announcement, directory, authority, CA
- X.509 authentication and certificates
- public key infrastructure (PKIX)

- Necessary for security mechanisms on various layers

Acks & Recommended Reading

Selected slides of this chapter courtesy of

- W. Stallings (L. Brown) with changes incorporated by myself

Additional readings will be made available via our blog

- Details on X.509 certificates (courtesy of S. Kent)
- Various documents to problems in PKIs

Recommended reading

- [KaPeSp2002] Charlie Kaufman, Radia Perlman, Mike Speciner: Network Security – Private Communication in a Public World, 2nd Edition, Prentice Hall, 2002, ISBN: 978-0-13-046019-6
- [Stallings2014] William Stallings, Network Security Essentials, 4th Edition, Prentice Hall, 2014, ISBN: 978-0-136-10805-4
- [Schäfer2003] G. Schäfer. Netzsicherheit - Algorithmische Grundlagen und Protokolle. dpunkt.verlag, 2003.

Copyright Notice

This document has been distributed by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically.

It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Contact

Prof. Dr.-Ing. Matthias Hollick
Department of Computer Science

SEEMOO
Mornewegstr. 32
64293 Darmstadt/Germany
matthias.hollick@seemoo.tu-darmstadt.de

Phone +49 6151 16-70920
Fax +49 6151 16-70921
www.seemoo.tu-darmstadt.de