Cryptography and Network Security Chapter 14

Fourth Edition
by William Stallings

Chapter 14 – Authentication Applications

In practice , the effectiveness of a countermeasure often depends on how it is used; the best safe in the world is worthless if no one remembers to close the door.

—Safe Computing in Information AGE, NRC

Authentication Applications

- will consider authentication functions
- developed to support application-level authentication & digital signatures
- will consider Kerberos – a private-key authentication service
- then X.509 - a public-key directory authentication service

Kerberos

- trusted key server system from MIT
- provides centralised private-key third-party authentication in a distributed network
  - allows users access to services distributed through network
  - without needing to trust all workstations
  - rather all trust a central authentication server
- two versions in use: 4 & 5

Kerberos Requirements

- its first report identified requirements as:
  - secure
  - reliable
  - transparent
  - scalable
- implemented using an authentication protocol based on Needham-Schroeder

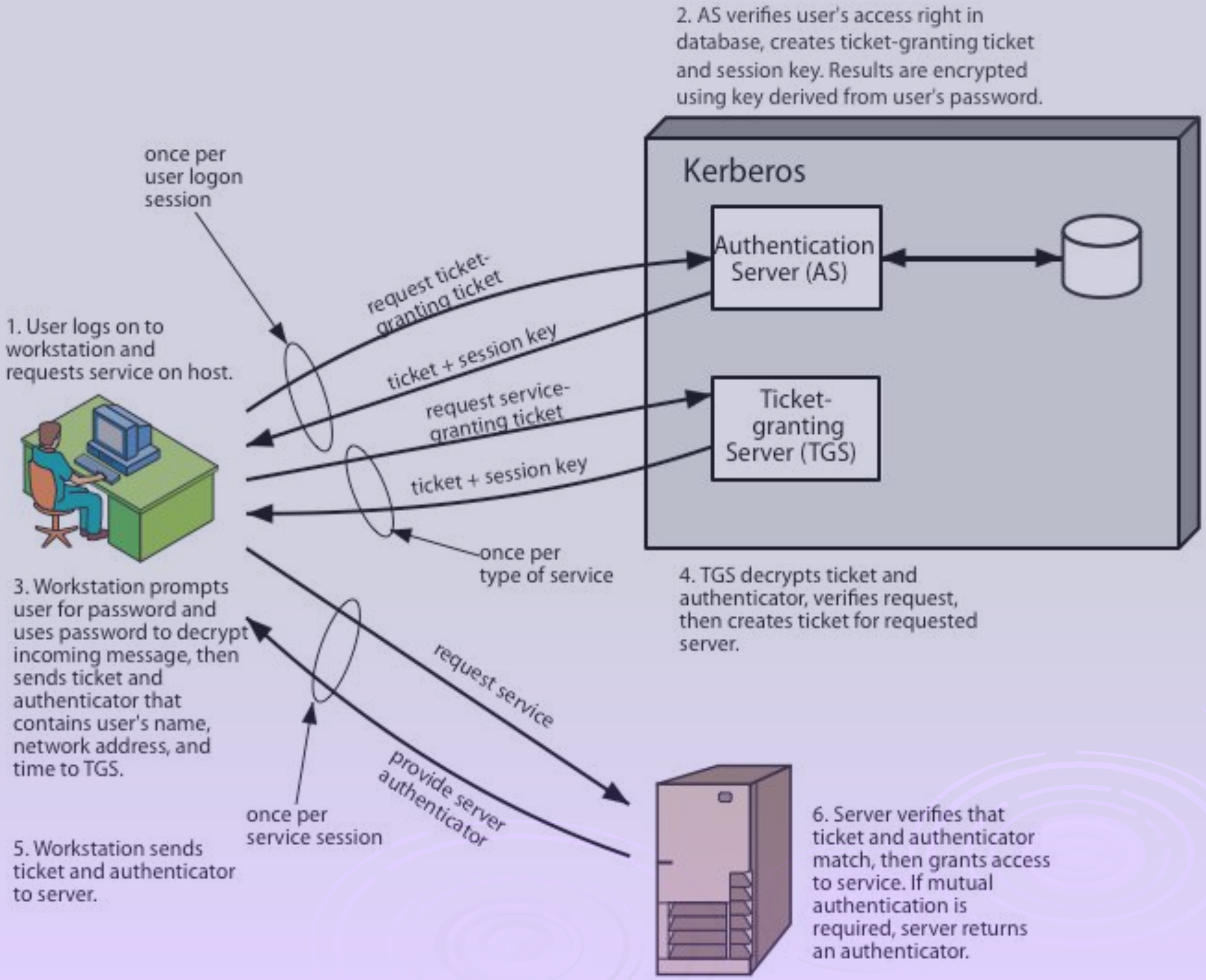
Kerberos v4 Overview

- a basic third-party authentication scheme
- have an Authentication Server (AS)
  - users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- have a Ticket Granting server (TGS)
  - users subsequently request access to other services from TGS on basis of users TGT

Kerberos v4 Dialogue

1. obtain ticket granting ticket from AS
   - once per session
2. obtain service granting ticket from TGT
   - for each distinct service required
3. client/server exchange to obtain service
   - on every service request

Kerberos 4 Overview



once per user logon session
request ticket-granting ticket
ticket + session key
request service-granting ticket
ticket + session key
once per type of service
Kerberos
Authentication Server (AS)
Ticket-granting Server (TGS)
request service
provide server authenticator
once per service session

1. User logs on to workstation and requests service on host.

2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.

4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.

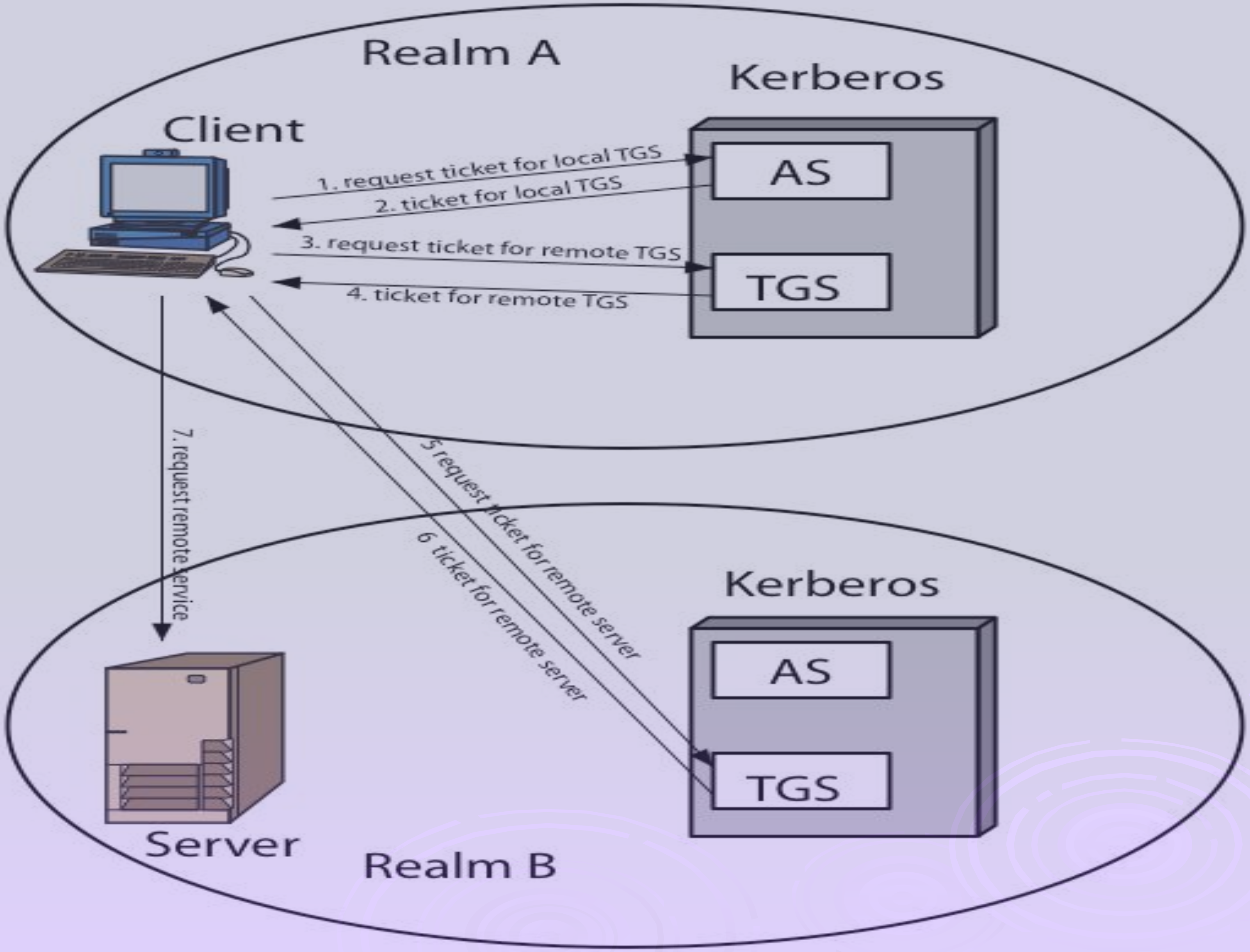
5. Workstation sends ticket and authenticator to server.

6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

Kerberos Realms

- a Kerberos environment consists of:
  - a Kerberos server
  - a number of clients, all registered with server
  - application servers, sharing keys with server
- this is termed a realm
  - typically a single administrative domain
- if have multiple realms, their Kerberos servers must share keys and trust

Kerberos Realms


Realm A
Kerberos
Client
AS
TGS
1. request ticket for local TGS
2. ticket for local TGS
3. request ticket for remote TGS
4. ticket for remote TGS
5 request ticket for remote server
6 ticket for remote server
7. request remote service
Kerberos
AS
TGS
Server
Realm B

Kerberos Version 5

- developed in mid 1990's
- specified as Internet standard RFC 1510
- provides improvements over v4
  - addresses environmental shortcomings
    - encryption alg, network protocol, byte order, ticket lifetime, authentication forwarding, interrealm auth
  - and technical deficiencies
    - double encryption, non-std mode of use, session keys, password attacks

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 in S/MIME, IP SEC, SSL/TLS, SET

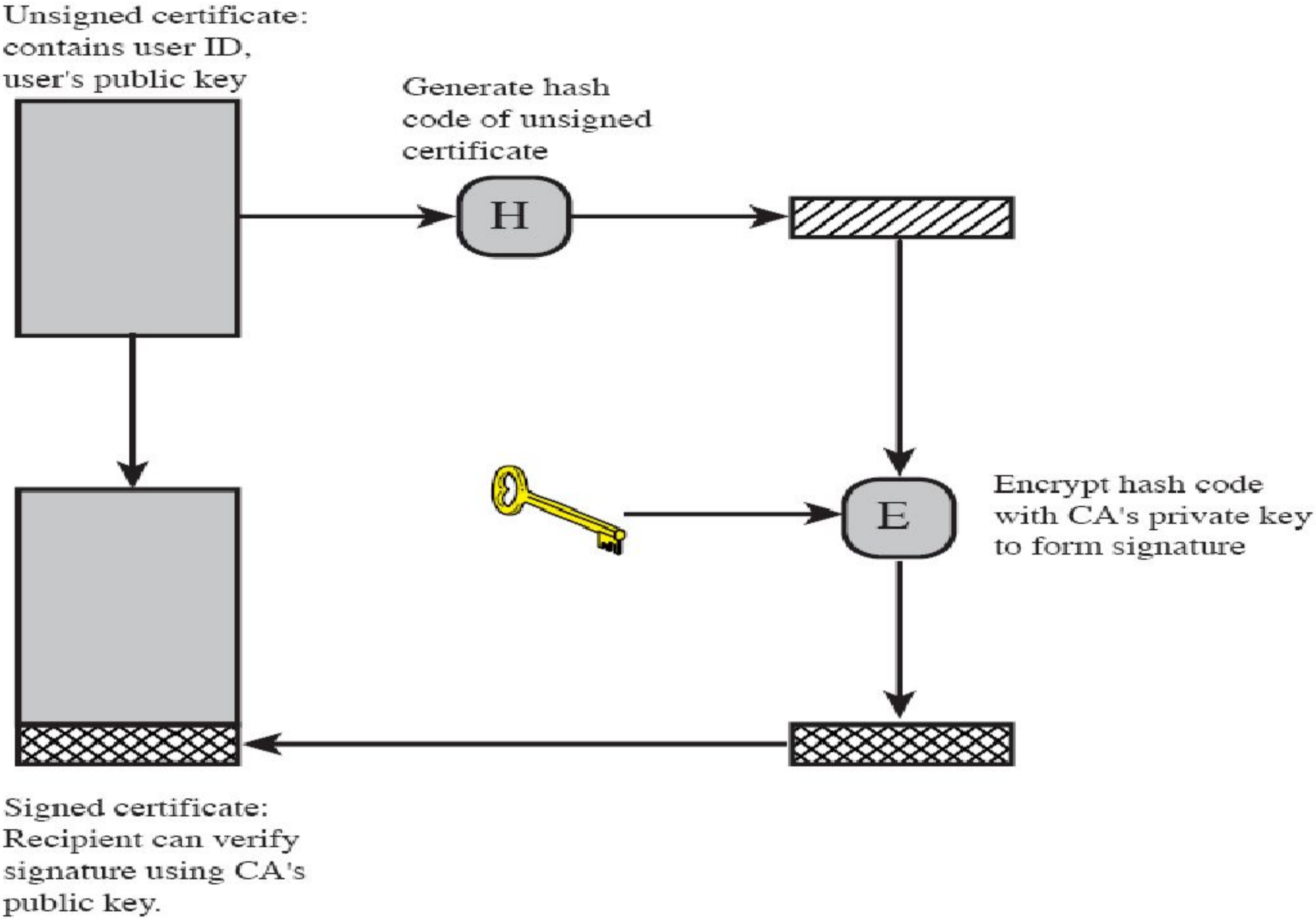


Figure 14.3 Public-Key Certificate Use

X.509 Certificates

- issued by a Certification Authority (CA), containing:
  - version (1, 2, or 3)
  - serial number (unique within CA) identifying certificate
  - signature algorithm identifier
  - issuer X.500 name (CA)
  - period of validity (from - to dates)
  - subject X.500 name (name of owner)
  - subject public-key info (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

What's Inside an X.509 Certificate?

The X.509 standard defines what information can go into a certificate, and describes how to write it down (the data format). All X.509 certificates have the following data, in addition to the signature:

Version

This identifies which version of the X.509 standard applies to this certificate, which affects what information can be specified in it. Thus far, three versions are defined.

Serial Number

The entity that created the certificate is responsible for assigning it a serial number to distinguish it from other certificates it issues. This information is used in numerous ways, for example when a certificate is revoked its serial number is placed in a Certificate Revocation List (CRL).

Signature Algorithm Identifier

This identifies the algorithm used by the CA to sign the certificate.

Issuer Name

The X.500 name of the entity that signed the certificate. This is normally a CA. Using this certificate implies trusting the entity that signed this certificate. (Note that in some cases, such as *root or top-level* CA certificates, the issuer signs its own certificate.)

Validity Period

Each certificate is valid only for a limited amount of time. This period is described by a start date and time and an end date and time, and can be as short as a few seconds or almost as long as a century. The validity period chosen depends on a number of factors, such as the strength of the private key used to sign the certificate or the amount one is willing to pay for a certificate. This is the expected period that entities can rely on the public value, if the associated private key has not been compromised.

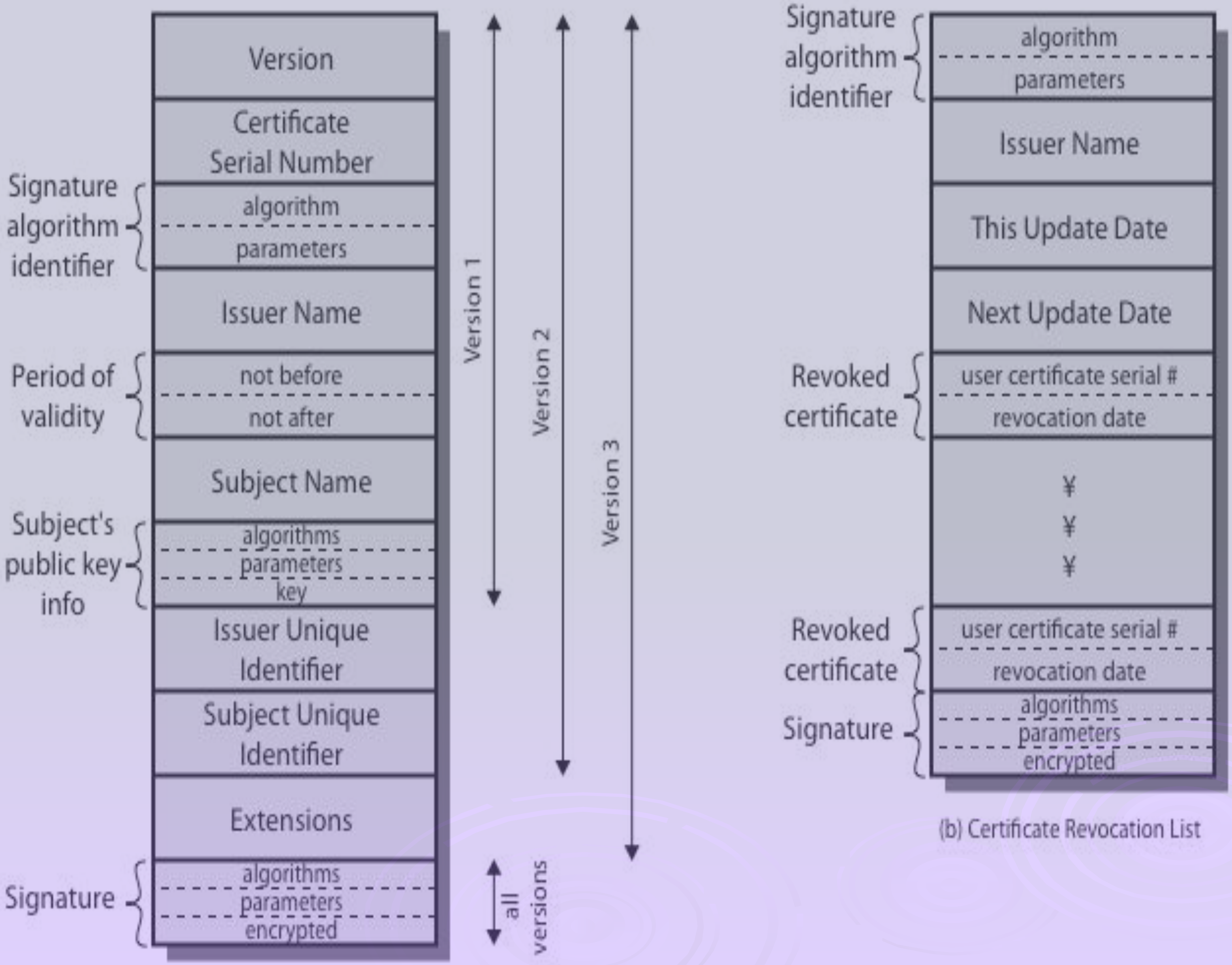
Subject Name

The name of the entity whose public key the certificate identifies. This name uses the X.500 standard, so it is intended to be unique across the Internet. This is the Distinguished Name (DN) of the entity, for example, CN=Java Duke, OU=Java Software Division, O=Sun Microsystems Inc, C=US (These refer to the subject's Common Name, Organizational Unit, Organization, and Country.)

Subject Public Key Information

This is the public key of the entity being named, together with an algorithm identifier which specifies which public key crypto system this key belongs to and any associated key parameters.

X.509 Certificates



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

Version 1

Version 2

Version 3

all versions

Signature algorithm identifier: algorithm; parameters

Issuer Name

This Update Date

Next Update Date

Revoked certificate: user certificate serial #; revocation date

¥
¥
¥

Revoked certificate: user certificate serial #; revocation date

Signature: algorithms; parameters; encrypted

(b) Certificate Revocation List

```
Certificate:
Data:
Version: 1 (0x0)
Serial Number: 7829 (0x1e95)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
        OU=Certification Services Division,
        CN=Thawte Server CA/emailAddress=server-certs@thawte.com
Validity
        Not Before: Jul 9 16:04:02 1998 GMT
        Not After : Jul 9 16:04:02 1999 GMT
Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
        OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit) Modulus (1024 bit):
00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb: 33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66: 70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b: c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3: d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
e8:35:1c:9e:27:52:7e:41:8f
Exponent: 65537 (0x10001)
Signature Algorithm: md5WithRSAEncryption
93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d: 92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:
19:f6:ad:ef:63:2f:92: ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67: d0:a2:40:
03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72: 0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:
85:a6:ef:19:d1: 5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7: 8f:0e:fc:ba:1f:
34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22: 68:9f
```

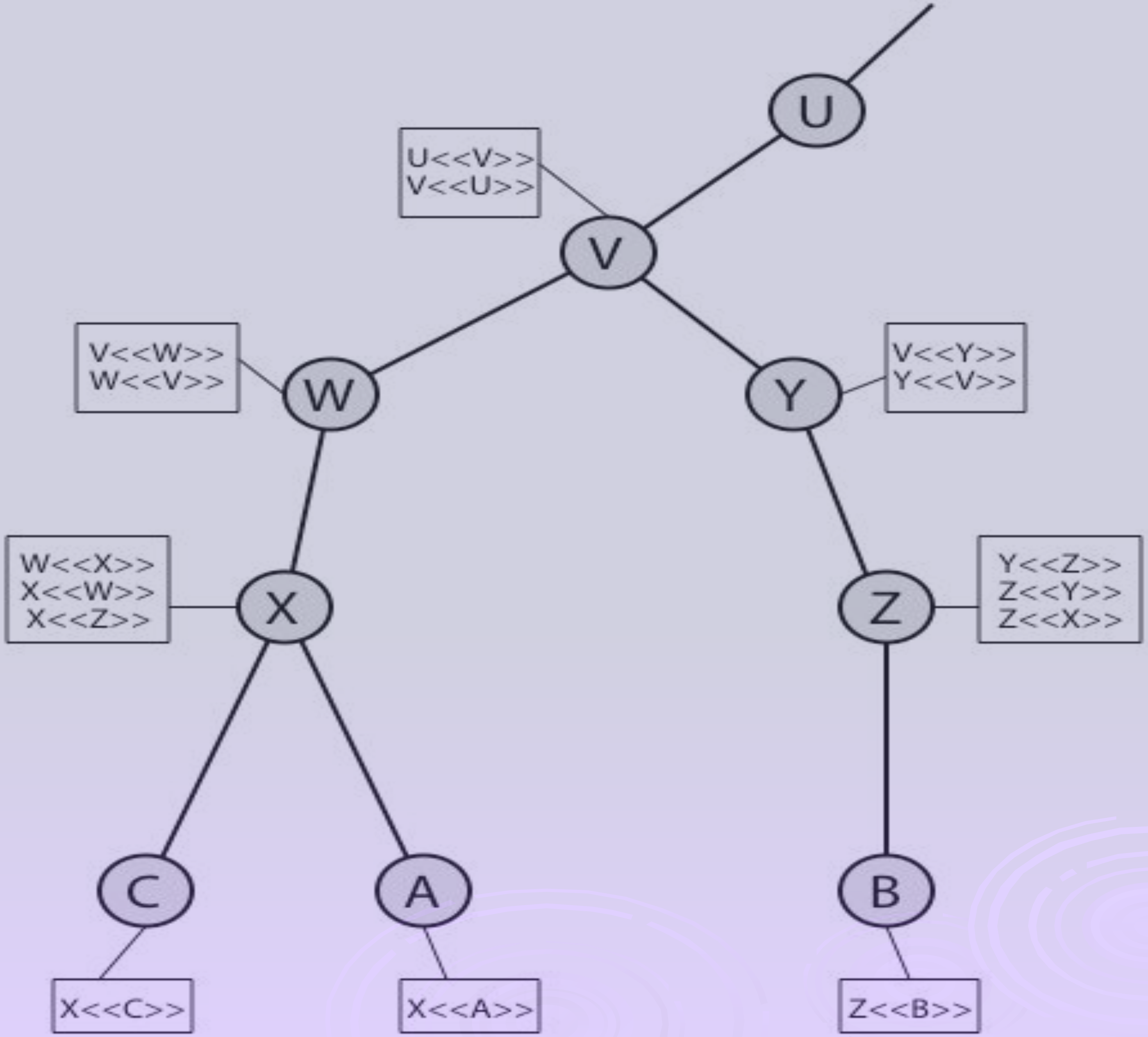
Obtaining a Certificate

- 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

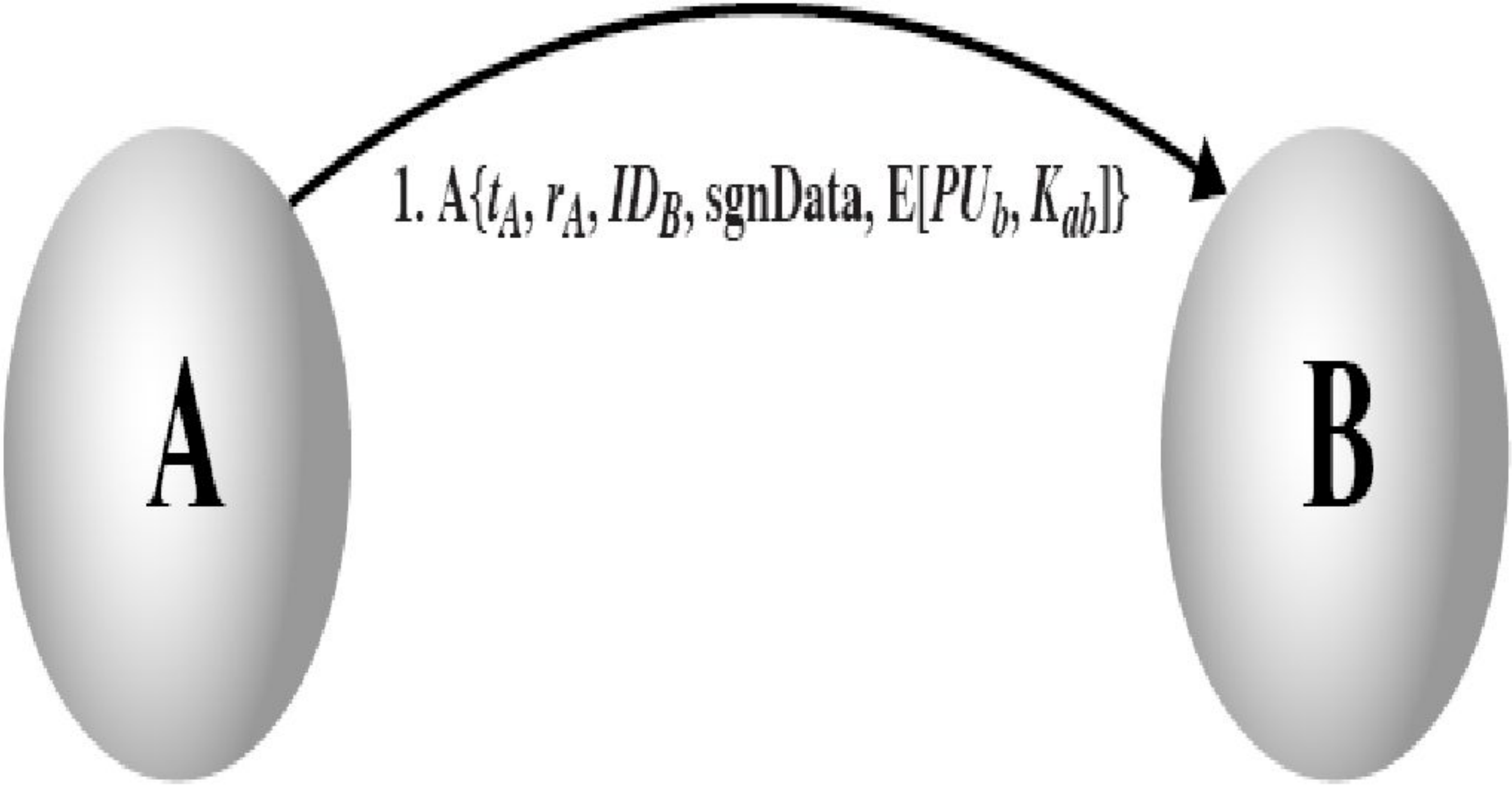
- certificates have a period of validity
- may need to revoke before expiry, eg:
  1. user's private key is compromised
  2. user is no longer certified by this CA
  3. CA's certificate is compromised
- CA’s maintain list of revoked certificates
  - the Certificate Revocation List (CRL)
- users should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

Authentication Procedures

- X.509 includes three alternative authentication procedures:
- One-Way Authentication
- Two-Way Authentication
- Three-Way Authentication
- all use public-key signatures

One-Way Authentication

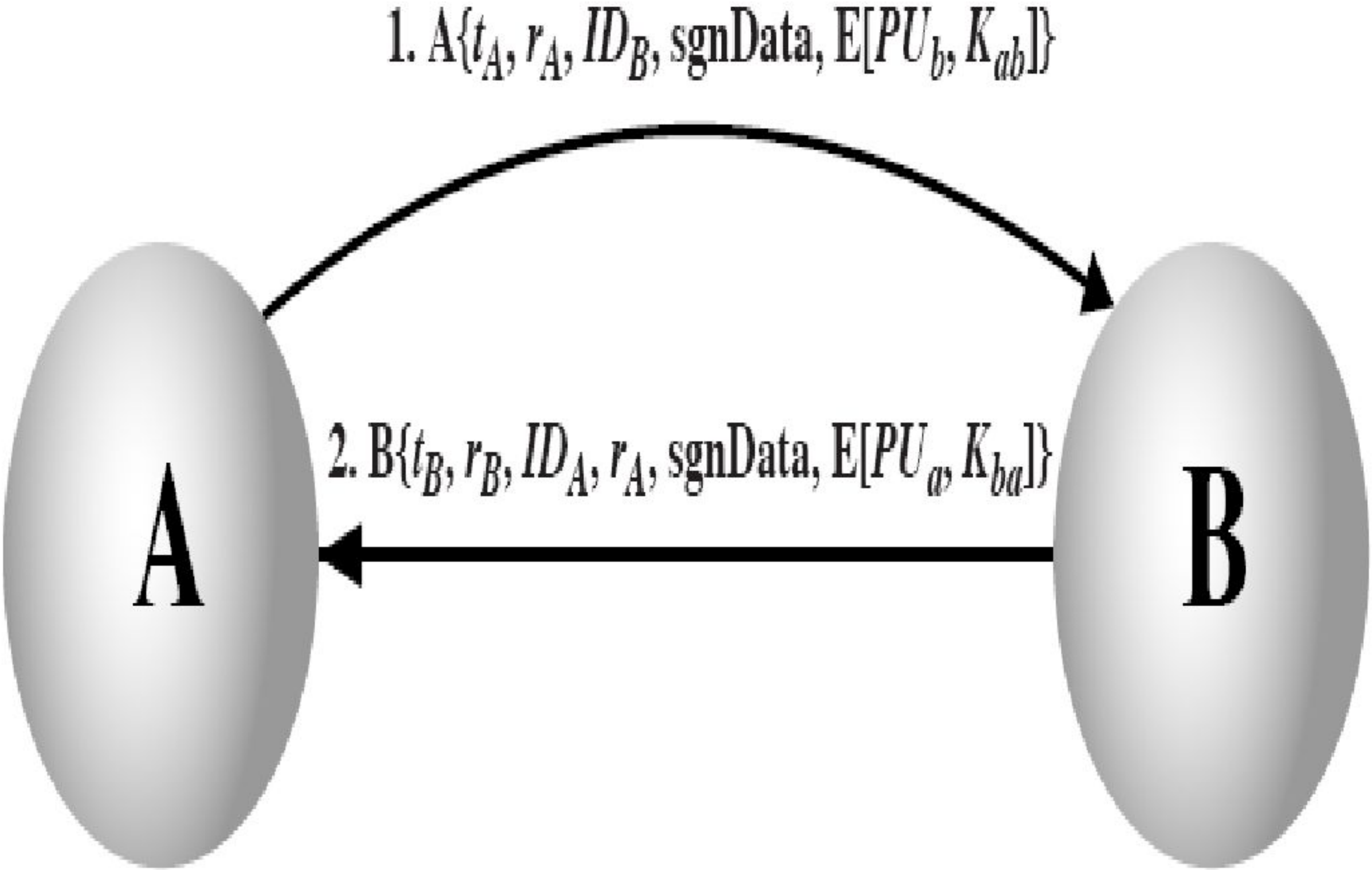
- 1 message (A->B) used to establish
  - the identity of A and that message is from A
  - message was intended for B
  - integrity & originality of message
- message must include timestamp, nonce, B's identity and is signed by A
- may include additional info for B
  - eg session key



(a) One-way authentication

Two-Way Authentication

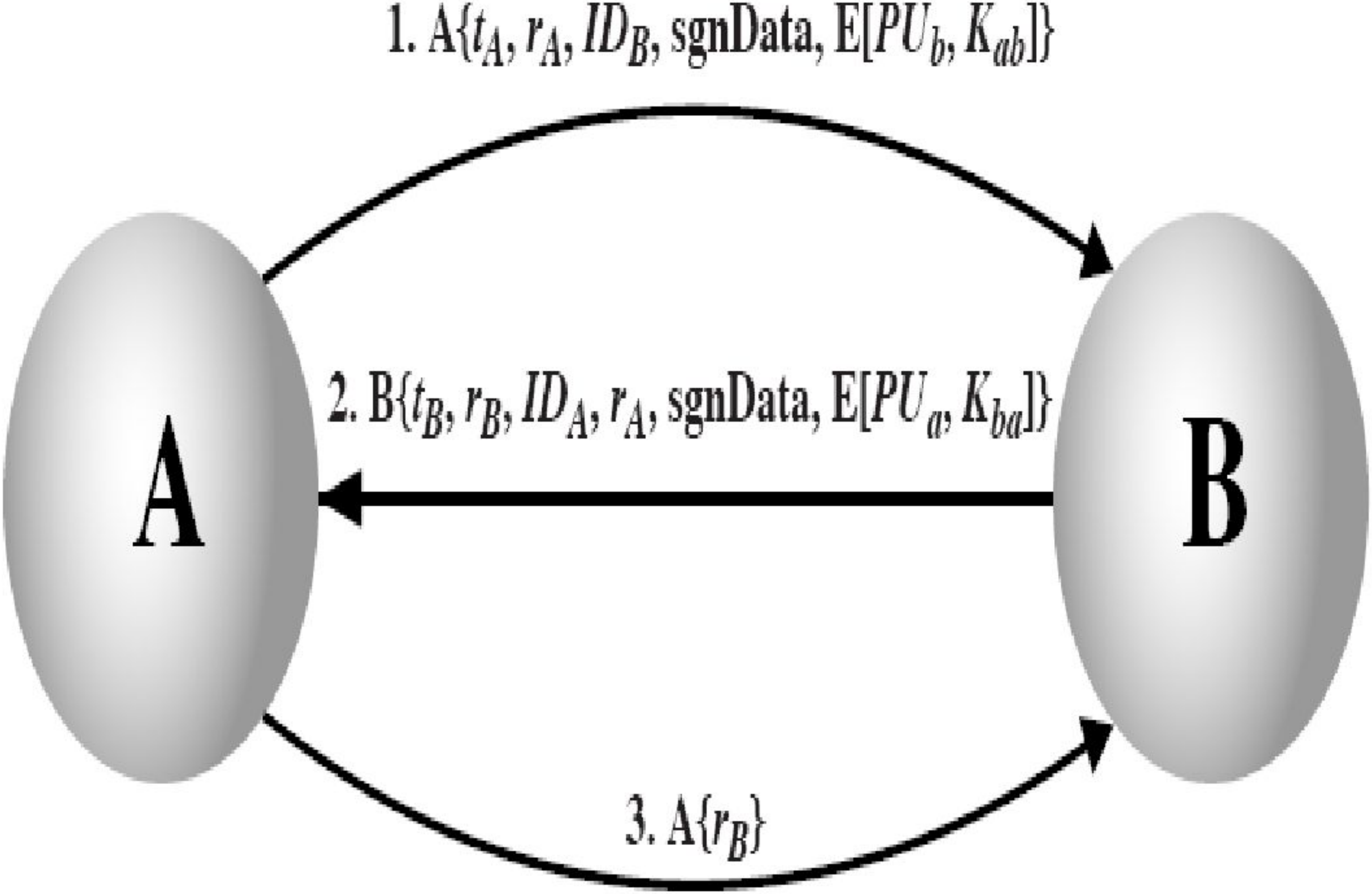
- 2 messages (A->B, B->A) which also establishes in addition:
  - the identity of B and that reply is from B
  - that reply is intended for A
  - integrity & originality of reply
- reply includes original nonce from A, also timestamp and nonce from B
- may include additional info for A


1. $A\{t_A, r_A, ID_B, \text{sgnData}, E[PU_b, K_{ab}]\}$
2. $B\{t_B, r_B, ID_A, r_A, \text{sgnData}, E[PU_a, K_{ba}]\}$
A
B

(b) Two-way authentication

Three-Way Authentication

- 3 messages (A->B, B->A, A->B) which enables above authentication without synchronized clocks
- has reply from A back to B containing signed copy of nonce from B
- means that timestamps need not be checked or relied upon


1. A$\{t_A, r_A, ID_B, \text{sgnData}, E[PU_b, K_{ab}]\}$
2. B$\{t_B, r_B, ID_A, r_A, \text{sgnData}, E[PU_a, K_{ba}]\}$
A
B
3. A$\{r_B\}$

(c) Three-way authentication

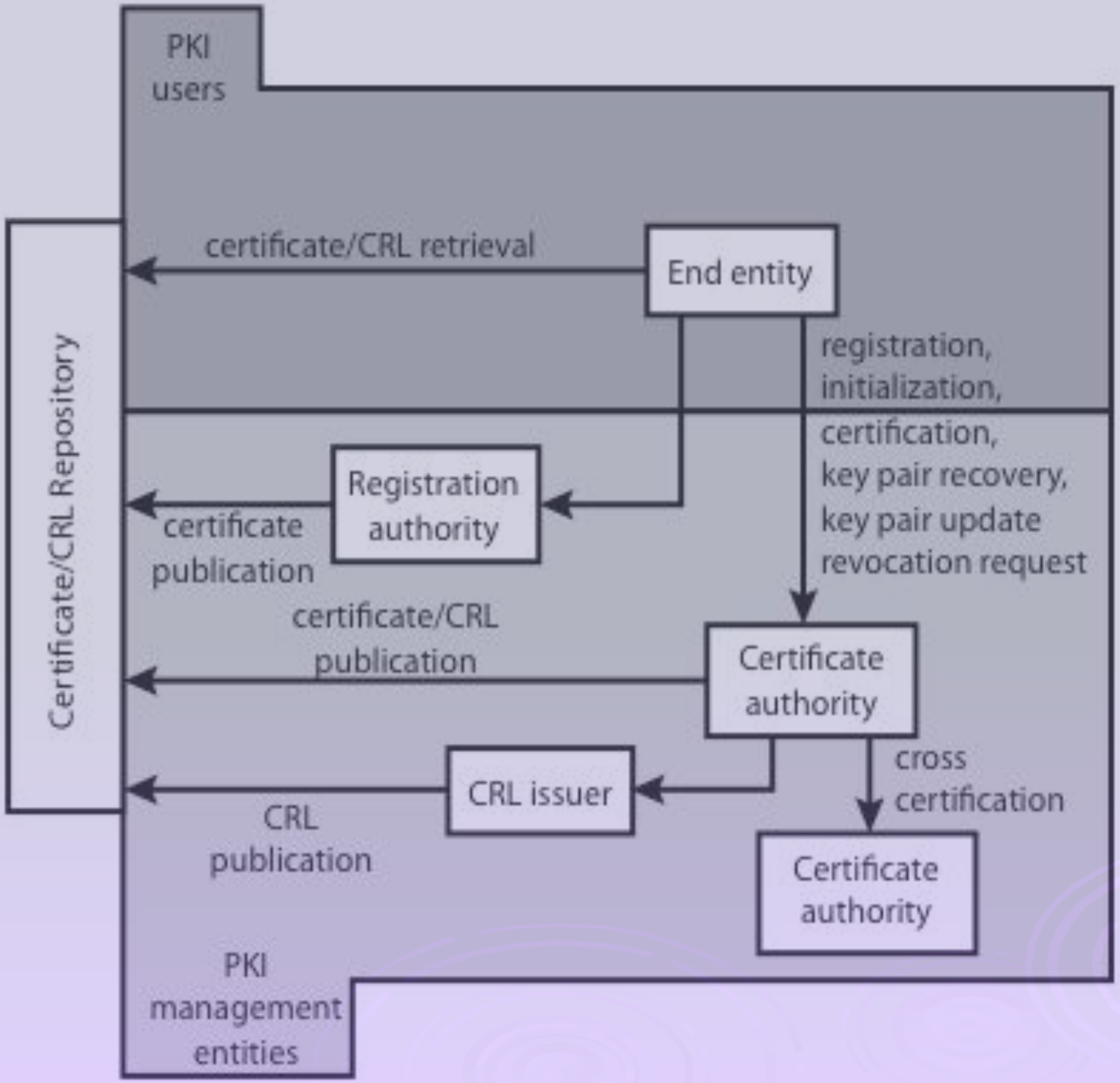
X.509 Version 3

- 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



Summary

- have considered:
  - Kerberos trusted key server system
  - X.509 authentication and certificates