Internet Authentication Applications

Bojan Božić

By: William Stallings and Lawrie Brown

Kerberos Overview

- Initially developed at MIT
- Software utility available in both the public domain and in commercially supported versions
- Issued as an Internet standard and is the defacto standard for remote authentication
- Overall scheme is that of a trusted third party authentication service
- Requires that a user prove his or her identity for each service invoked and requires servers to prove their identity to clients

Kerberos Protocol

Involves clients, application servers, and a Kerberos server

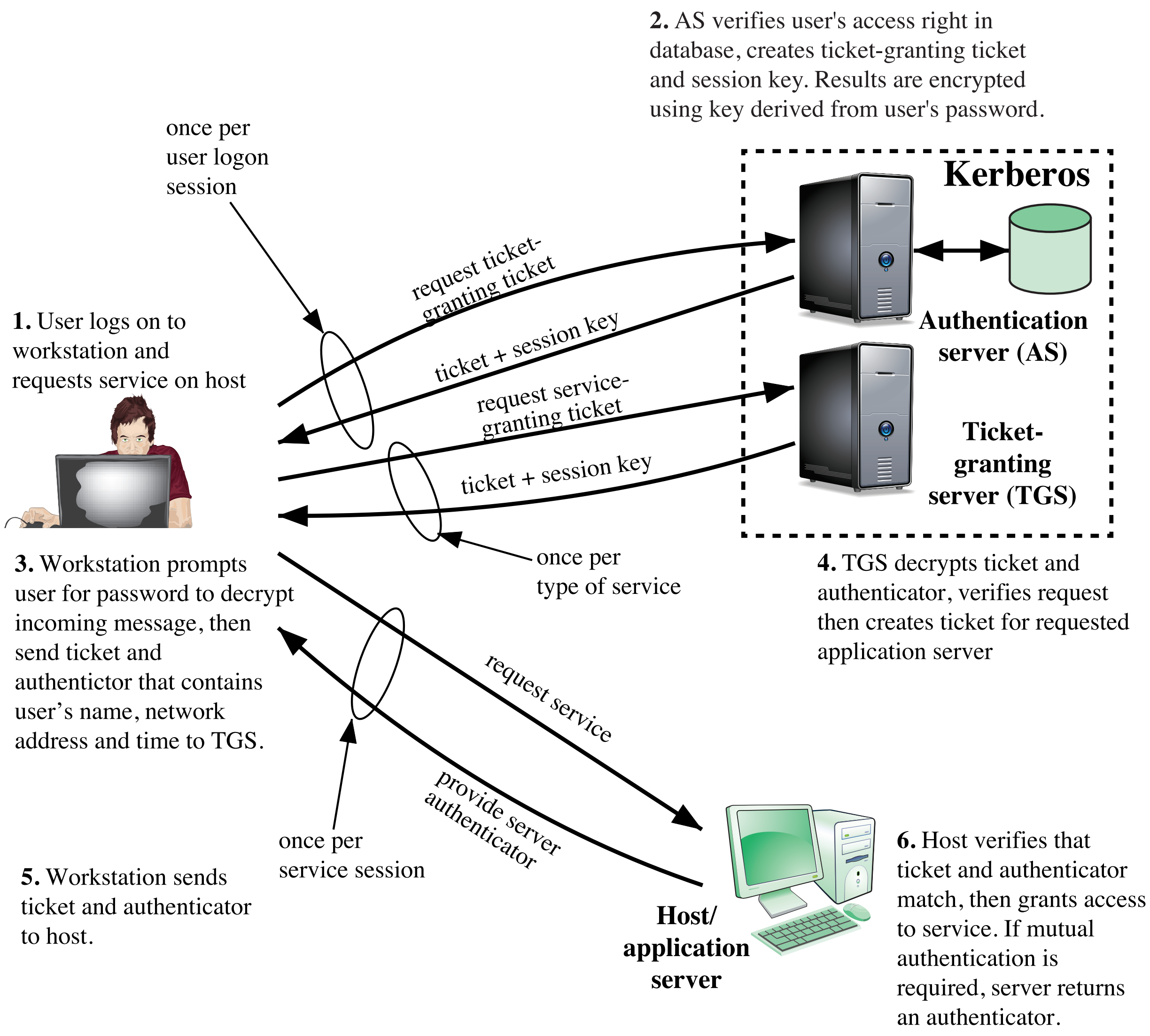
- Designed to counter a variety of threats to the security of a client/server dialogue
- Obvious security risk is impersonation
- Servers must be able to confirm the identities of clients who request service

Use an Authentication Server (AS)

- User initially negotiates with AS for identity verification
- AS verifies identity and then passes information on to an application server which will then accept service requests from the client

Need to find a way to do this in a secure way

- If client sends user's password to the AS over the network an opponent could observe the password
- An opponent could impersonate the AS and send a false validation



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 to decrypt incoming message, then send ticket and authentictor that contains user’s name, network address and time to TGS.

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

5. Workstation sends ticket and authenticator to host.

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

Figure 23.1 Overview of Kerberos

Kerberos Realms

- A Kerberos environment consists of:
  - A Kerberos server
  - A number of clients, all registered with server
  - A number of application servers, sharing keys with server
- This is referred to as a realm
  - Networks of clients and servers under different administrative organizations generally constitute different realms
- If multiple realms:
  - Their Kerberos servers must share a secret key and trust the Kerberos server in the other realm to authenticate its users
  - Participating servers in the second realm must also be willing to trust the Kerberos server in the first realm

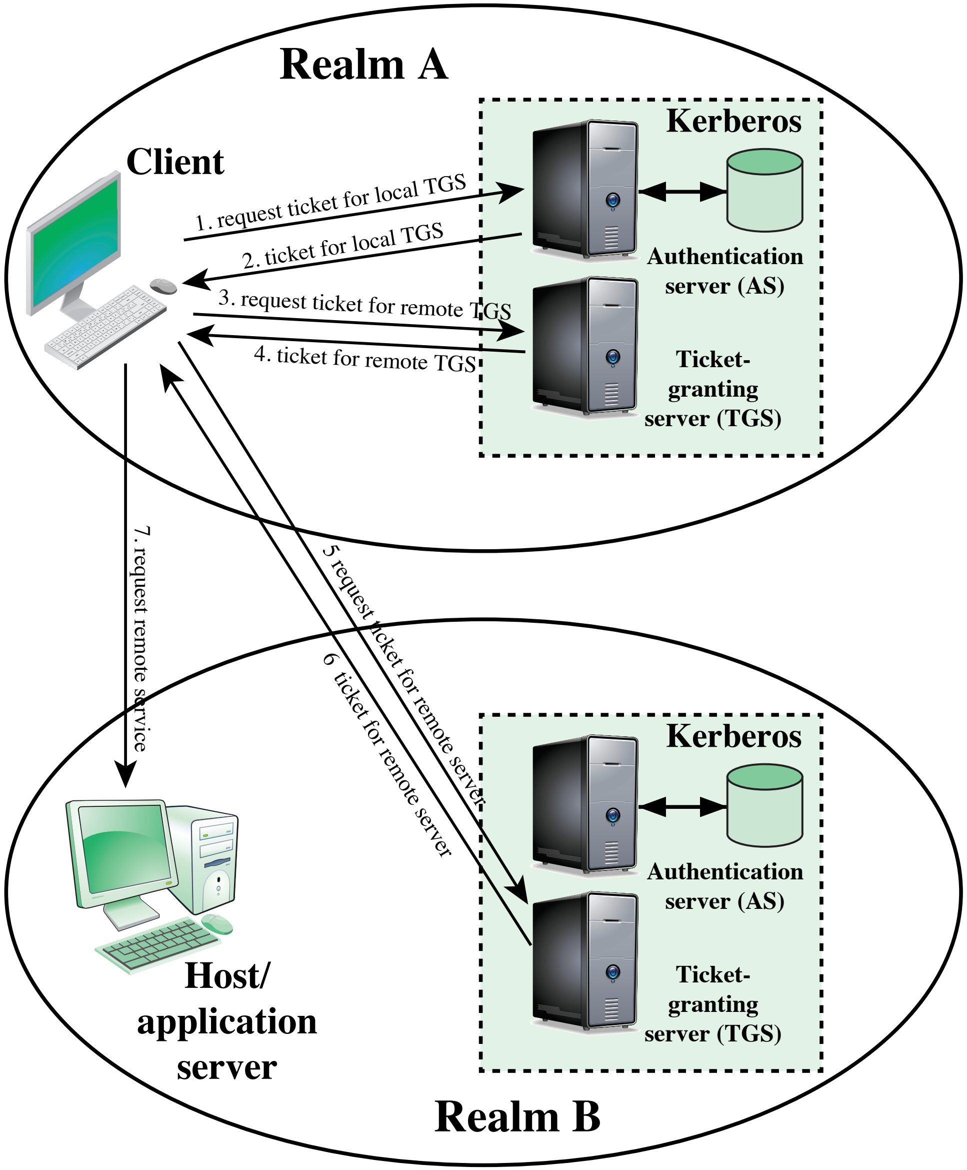
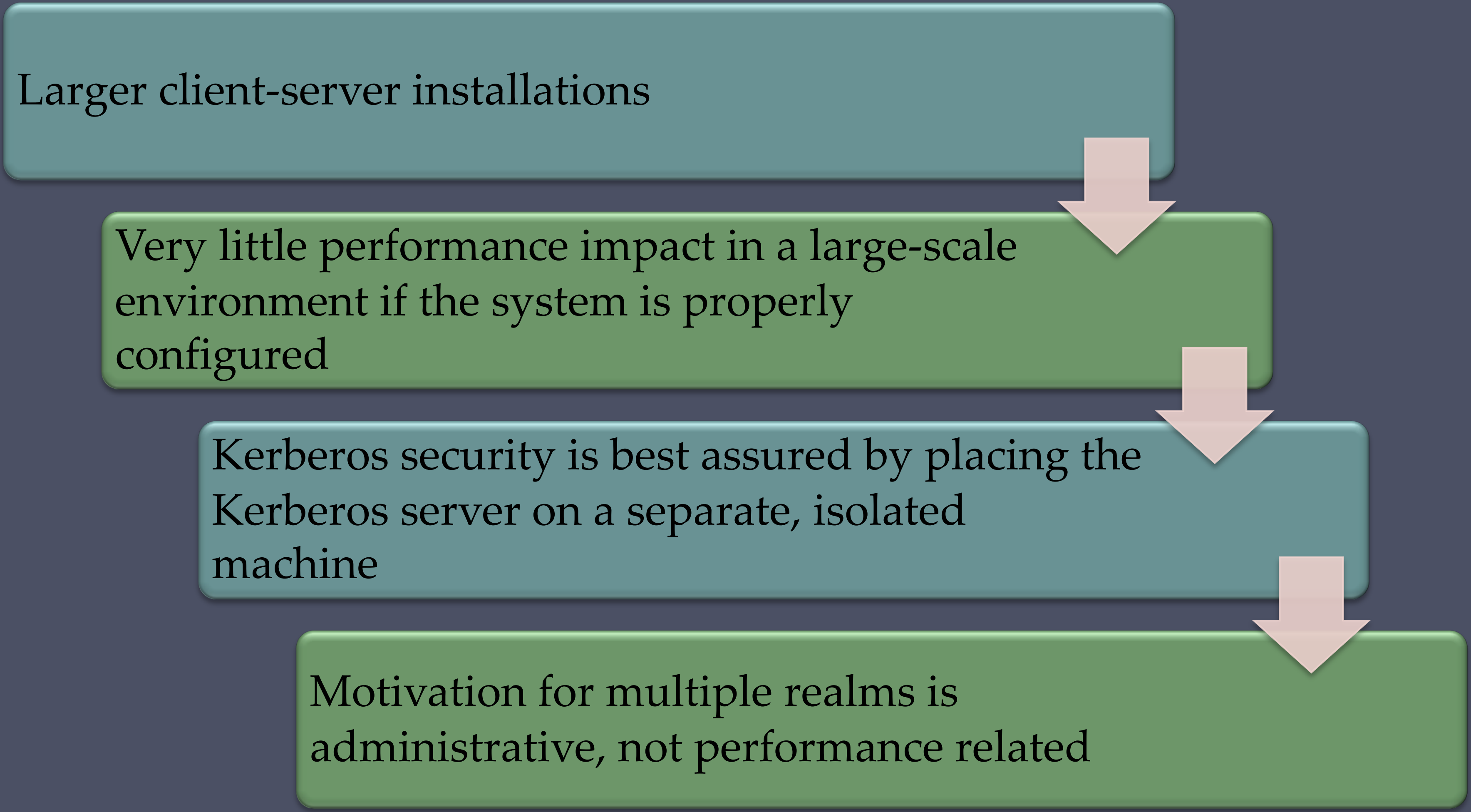

Realm A
Client
Kerberos
1. request ticket for local TGS
2. ticket for local TGS
3. request ticket for remote TGS
4. ticket for remote TGS
Authentication server (AS)
Ticket-granting server (TGS)
7. request remote service
5 request ticket for remote server
6 ticket for remote server
Kerberos
Authentication server (AS)
Ticket-granting server (TGS)
Host/ application server
Realm B

Figure 23.2 Request for Service in Another Realm

Kerberos Versions 4 and 5

- The first version of Kerberos that was widely used was version 4, published in the late 1980s
- Improvements found in version 5:
  - An encrypted message is tagged with an encryption algorithm identifier
    - This enables users to configure Kerberos to use an algorithm other than DES
  - Supports authentication forwarding
    - Enables a client to access a server and have that server access another server on behalf of the client
    - Supports a method for interrealm authentication that requires fewer secure key exchanges than in version 4

Kerberos Performance Issues



Larger client-server installations

Very little performance impact in a large-scale environment if the system is properly configured

Kerberos security is best assured by placing the Kerberos server on a separate, isolated machine

Motivation for multiple realms is administrative, not performance related

Certificate Authority (CA)

Certificate consists of:

- A public key with the identity of the key's owner
- Signed by a trusted third party
- Typically the third party is a CA that is trusted by the user community (such as a government agency, telecommunications company, financial institution, or other trusted peak organization)

User can present his or her public key to the authority in a secure manner and obtain a certificate

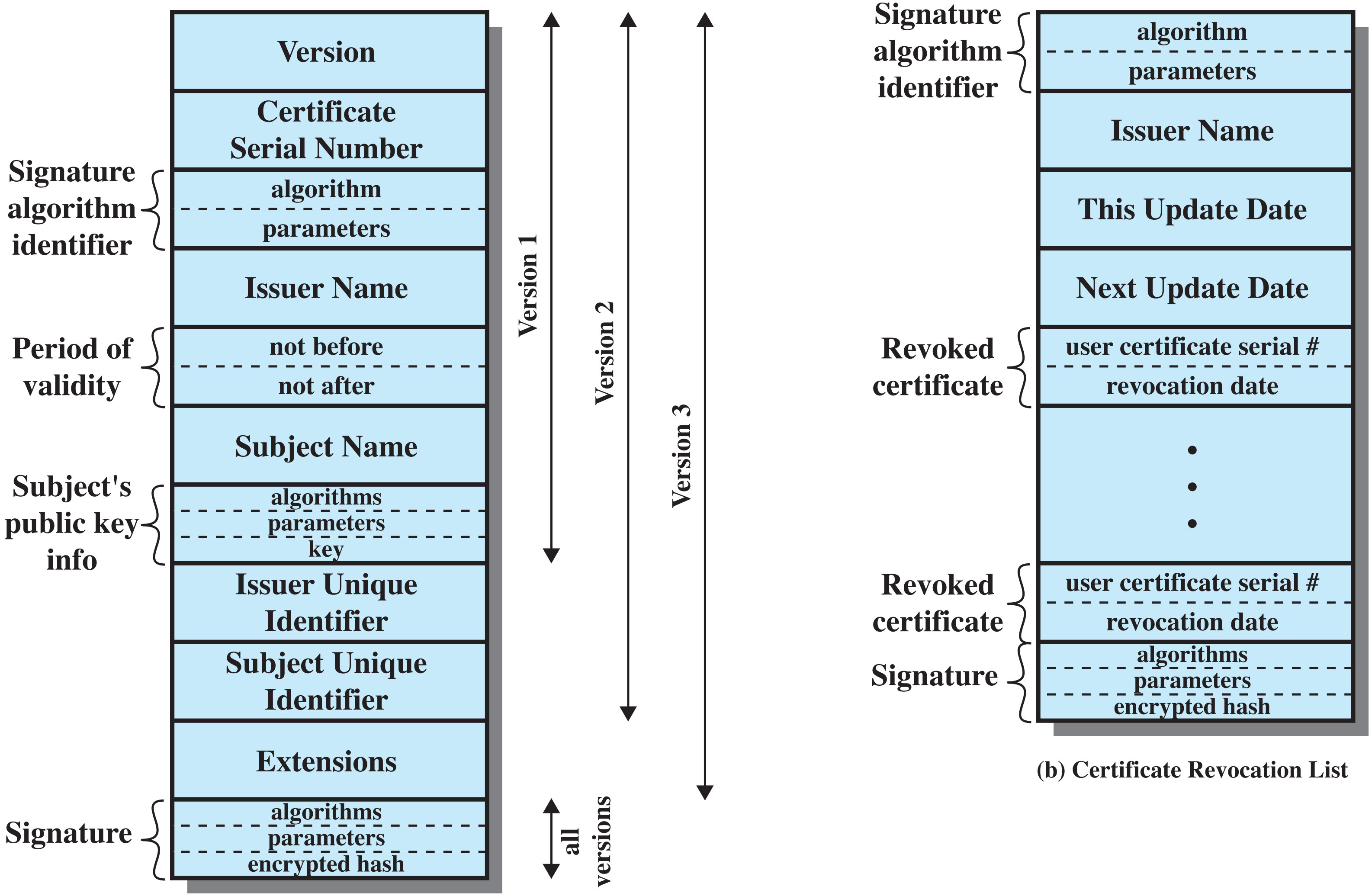
- User can then publish the certificate or send it to others
- Anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature

X.509

- Specified in RFC 5280
- The most widely accepted format for public-key certificates
- Certificates are used in most network security applications, including:
  - IP security (IPSEC)
  - Secure sockets layer (SSL)
  - Secure electronic transactions (SET)
  - S/MIME
  - eBusiness applications

A number of specialized variants also exist, distinguished by particular element values or the presence of certain extensions:

- Conventional (long-lived) certificates
  - CA and "end user" certificates
  - Typically issued for validity periods of months to years
- Short-lived certificates
  - Used to provide authentication for applications such as grid computing, while avoiding some of the overheads and limitations of conventional certificates
  - They have validity periods of hours to days, which limits the period of misuse if compromised
  - Because they are usually not issued by recognized CA's there are issues with verifying them outside their issuing organization
- Proxy certificates
  - Widely used to provide authentication for applications such as grid computing, while addressing some of the limitations of short-lived certificates
  - Defined in RFC 3820
  - Identified by the presence of the "proxy certificate" extension
  - They allow an "end user" certificate to sign another certificate
  - Allow a user to easily create a credential to access resources in some environment, without needing to provide their full certificate and right
- Attribute certificates
  - Defined in RFC 5755
  - Use a different certificate format to link a user's identity to a set of attributes that are typically used for authorization and access control
  - A user may have a number of different attribute certificates, with different set of attributes for different purposes
  - Defined in an "Attributes" extension



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

Version 1

Version 2

Version 3

all versions

(a) X.509 Certificate

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 hash

(b) Certificate Revocation List

Figure 23.3 X.509 Formats

Public-Key Infrastructure (PKI)

- The set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography
- Developed to enable secure, convenient, and efficient acquisition of public keys
- "Trust store"
  - A list of CA's and their public keys

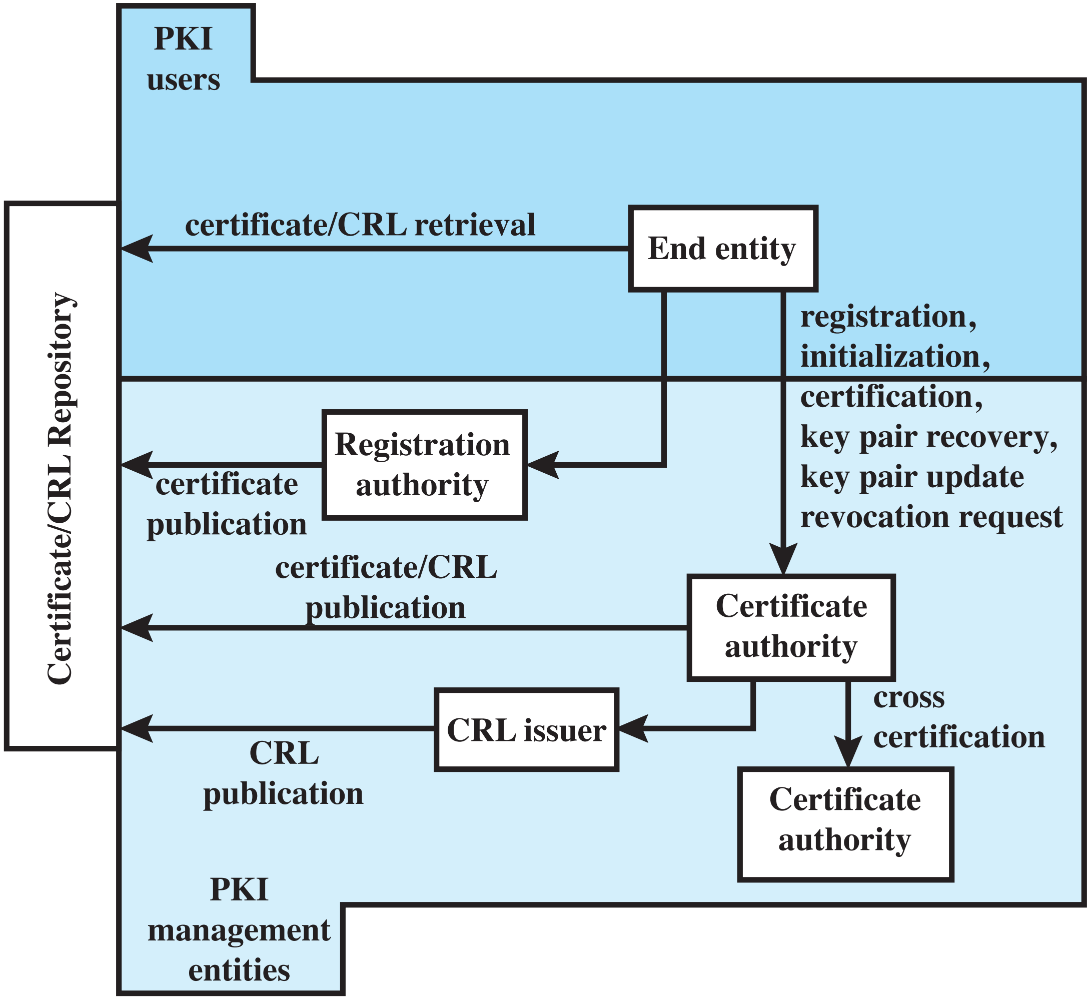


Figure 23.4 PKIX Architectural Model

Summary

- Kerberos
  - The Kerberos Protocol
  - Kerberos realms and multiple Kerberi
  - Version 4 and Version 5
  - Performance issues

- X.509
- Public Key infrastructure
  - Public Key infrastructure X.509 (PKIX)