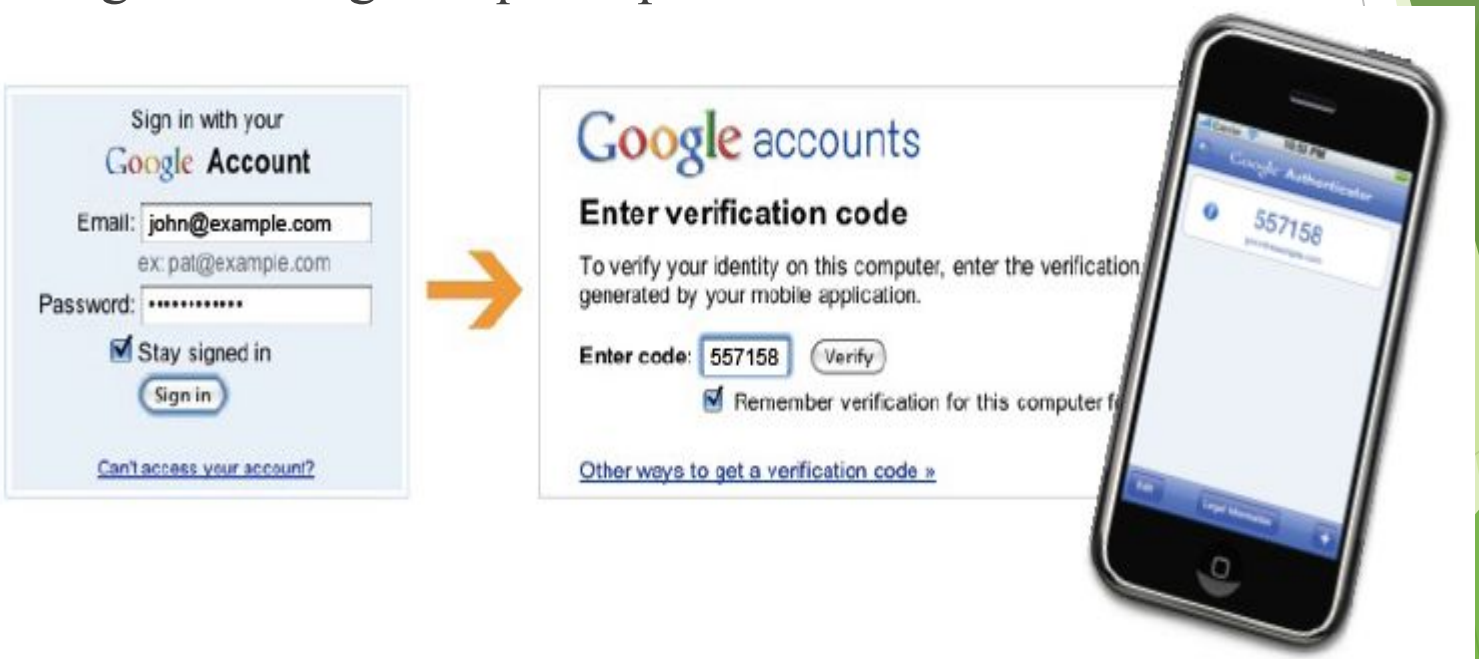
Authentication Examples

Dr. Nausheen Shoaib

Example: Google 2-Step Verification

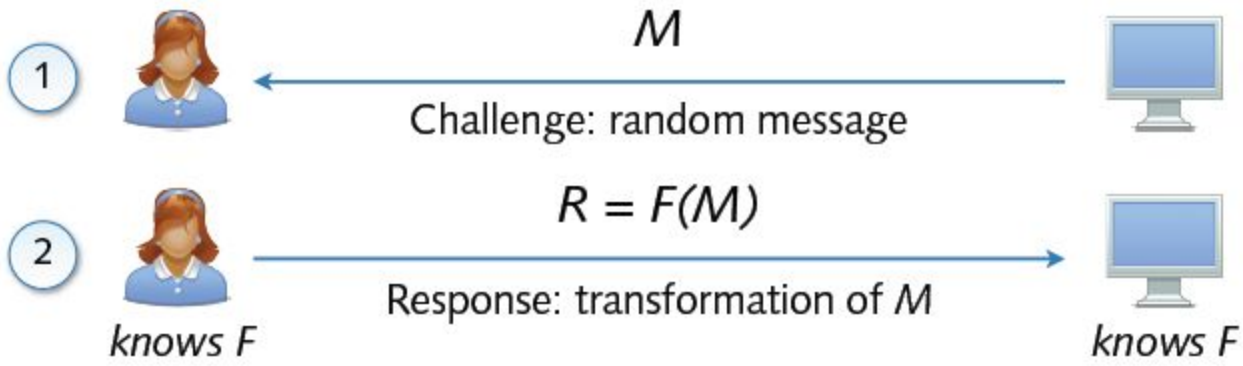
- **Security system by Google similar to SecurID**
  - Factors: knowledge (password) and ownership (phone)
  - Authentication code computed on mobile phone
  - Login at Google requires password and current code



https://blog.duosecurity.com/2013/02/bypassing-googles-two-factor-authentication/

Challenge-Reponse

- **Generic protocol scheme for authentication**
  - System and user share a secret function F



- **Advantages over naive authentication methods**
  - Secret, e.g. password, is never transmitted in cleartext
  - Replay attacks against authentication not possible

Challenge-Response (con't)

- **Secret function often parameterized by password**
  - F = H(M + P) hash function H and password P
  - F = E_P(M) encryption function E and password P
  - Hard to deduce P if F is cryptographically strong

- **Several methods related to challenge-response scheme**
  - One-time passwords
    - = challenge (index of password); response (password)
  - SecurID / Google 2-step
    - = challenge (current time); response (authentication code)

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

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

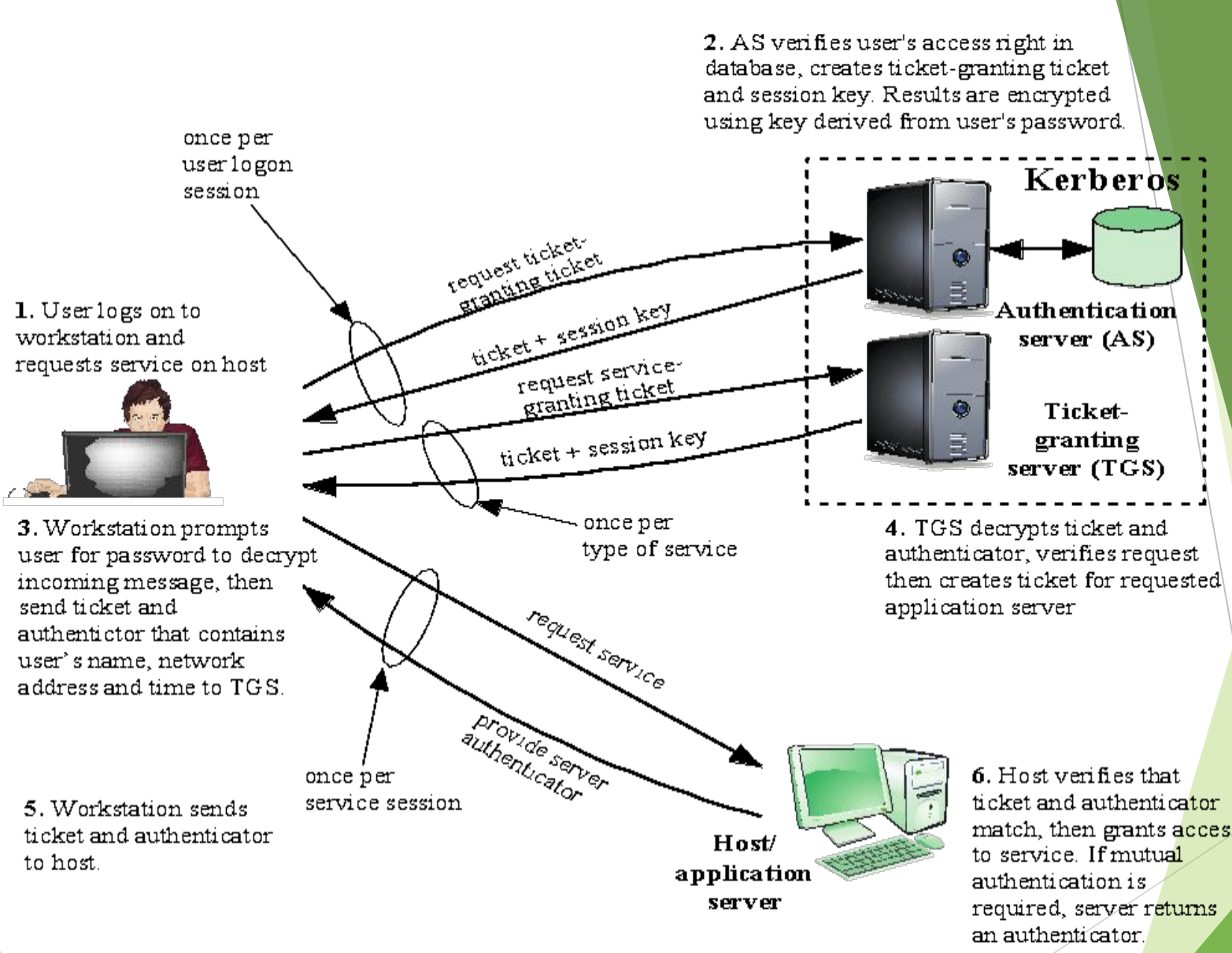


Figure 23.1 Overview of Kerberos

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

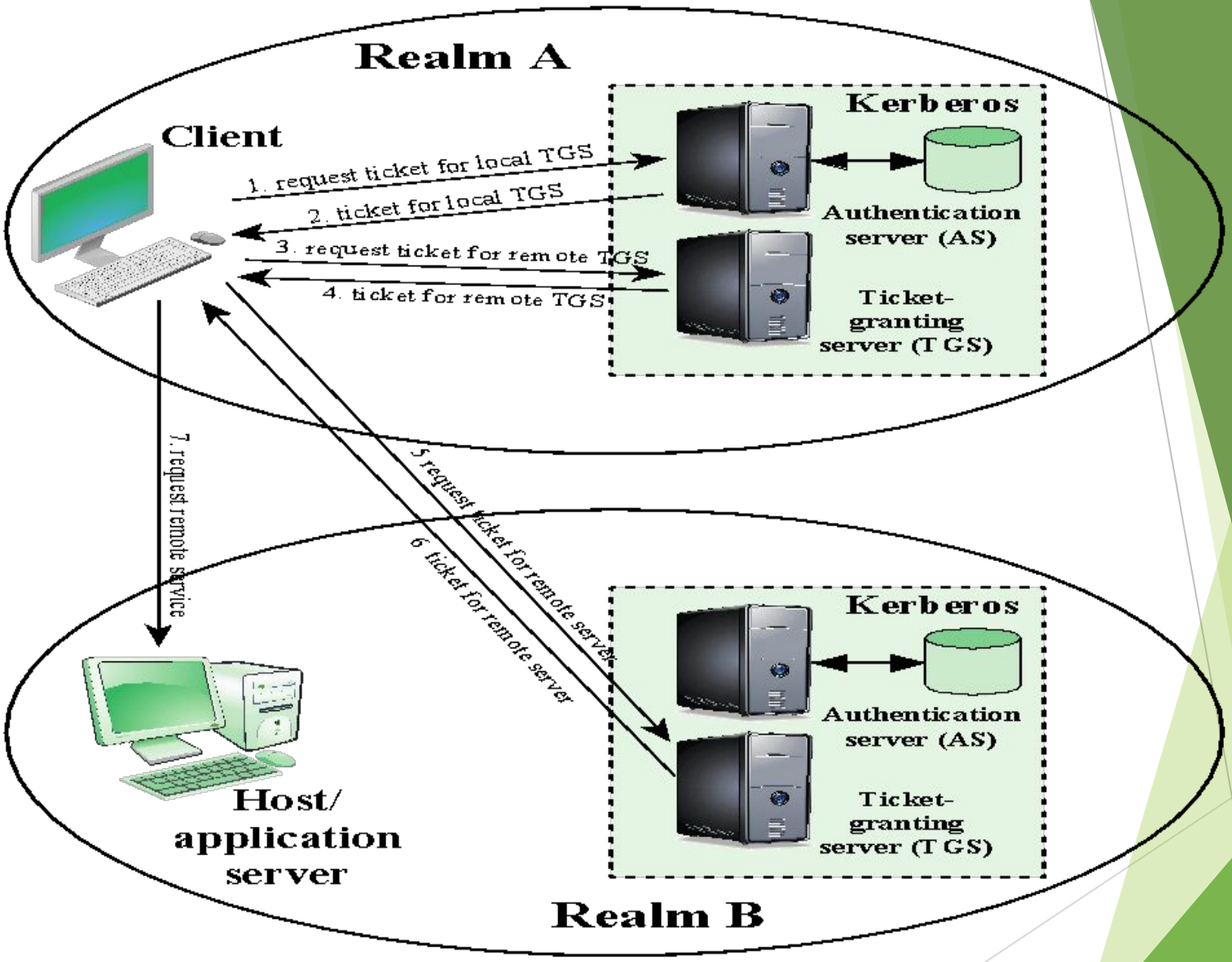


Figure 23.2 Request for Service in Another Realm

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

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

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

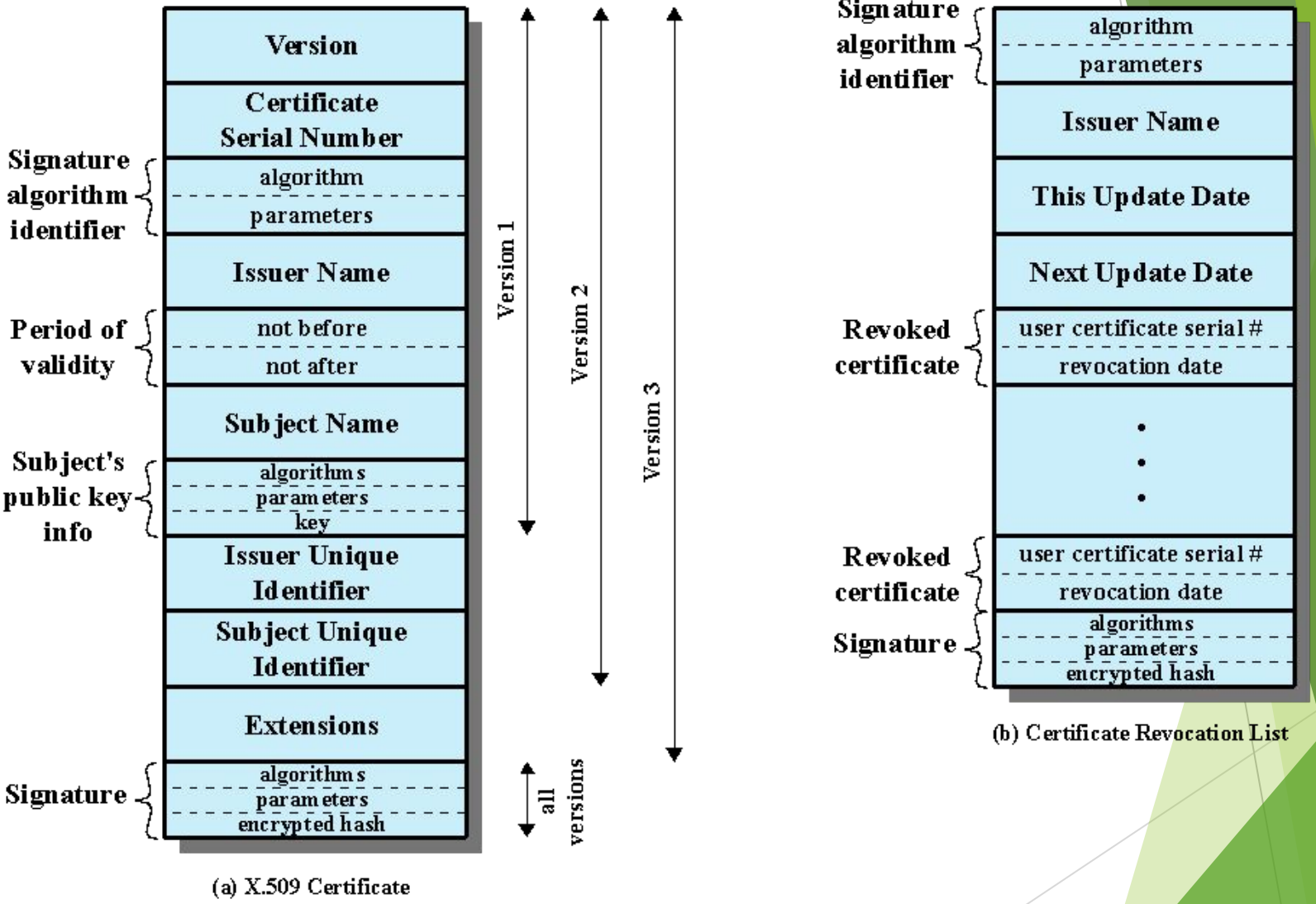


Figure 23.3 X.509 Formats