Digital Certificates and X.509 Authentication Service

Digital Certificates

- A digital certificate is:
  - An assertion
  - Digitally signed by a "certificate authority"
- An assertion
  - Can be anything
  - Usually an identity assertion
  - Can also be a list of authorizations
Public-Key Certificates

- reliable distribution of public-keys
  - public-key encryption
    - sender needs public key of receiver
  - public-key digital signatures
    - receiver needs public key of sender
  - public-key key agreement
    - both need each other’s public keys

Digital Certificates

- A certificate authority (CA) is
  - Someone who signs certificates
  - Has a “known” public key
  - Is “famous” enough for this to be useful
- Thus, a certificate is
  - A cryptographic proof that the CA believes the assertions
X.509 Certificate Authority Scope

A CA can vary dramatically in scope.

- At the large end are commercial CAs like Thawte, Verisign, Belsign, GTE Cybertrust or others.
  - These commercial CAs issue certificates to millions of users.
- At the smaller end are CAs operated by departments within a company:
  - These CAs issue certificates to a small number of users.
  - These smaller CAs may be intermediate CAs whose certificates are signed by higher-level CAs inside the organization.

X.509 Authentication Service Introduction

- ITU-T X.509:
  - Part of X.500 Directory Services
  - Issued in 1988; revised in 1993 and 1995
  - Defines a framework for authentication service using the X.500 directory
    - Repository of public-key certificates
  - Based on use of public-key cryptography and digital signatures
  - Recommends use of RSA
X.500 Directory

- X.500 Directory
  - Repository of public-key certificates
    - Public key of user
    - Signed with private key of trusted third party
  - Server (or set of servers) that maintain a user information database
    - Mapping from user name to network address
    - Other user attributes and information

Public-key Certificates

- Associated with user
  - Created by trusted third party
    - Certificate authority (CA)
    - Placed in directory by CA or by the user
  - Directory server
    - location for certificate access
    - does not create the certificates
Example of X.509 Certificate

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,

Subject Public Key Info:

Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
Exponent: 010001 (0x10001)

Signature Algorithm: md5WithRSAEncryption
X.509 Certificate Format

- The general format for a certificate is:
  - Version \( V \)
  - Serial number \( SN \)
  - Signature algorithm identifier \( AI \)
  - Issuer Name \( CA \)
  - Period of Validity \( TA \)
  - Subject Name \( A \)
  - Subject's Public-key Information \( Ap \)
  - Issuer Unique Identifier (added in Version 2)
  - Subject Unique Identifier (added in Version 2)
  - Extensions (added in Version 3)
  - Signature

X.509 Standard Notation

- User certificates generated by a CA use the following standard notation:
  \[ CA{<<A>>} = CA \{ V, SN, AI, CA, TA, A, Ap \} \]

  where
  \[ Y{<<X>>} = \] the certificate of user X issued by the certification authority Y
  \[ Y\{I\} = \] the signing of I by Y consisting of I with an encrypted hash code appended.
X.509: Obtaining A User Certificate

- User certificates generated by a CA have the following characteristics:
  - Any user with access to the public key of the CA can recover the user public key that was certified.
  - No party other than the CA can modify the certificate without being detected.
  - Since they are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them.

X.509: CA Trust Issues

- If all users subscribe to the same CA, then there is a common trust of that CA.
  - All user certificates can be placed in the directory for access by all users.
  - Any user can transmit his/her certificate directly to other users.
  - Once B is in possession of A’s certificate, B has confidence that:
    - Messages it encrypts will be secure.
    - Messages signed with A’s private key are unforgeable.
X.509: Multiple CAs

- Large User Community
  - Not Practical to Support All Users
  - More Practical to Have Multiple CAs
  - Each CA Provides Its Public Key to a Smaller User Group

X.509 Multiple CAs: Problem

- Consider this Scenario ...
  - User A obtained A’s certificate from CA X1.
  - User B obtained B’s certificate from CA X2.
  - If A does not know X2’s public key, B’s certificate is useless.
    - A can read B’s certificate
    - A cannot verify the signature
**X.509 Multiple CAs: Solution**

- Solution: CAs X1 and X2 exchange public keys

- Now...
  - A gets X2's certificate signed by X1
  - A gets B's certificate signed by X2
  - Now, A has trusted copy of X2's public key
    - Verifies the signature
    - Obtains B's public key

---

**X.509: CA Hierarchy Example**
X.509: Certificate Revocation

- Certificates have a period of validity, a *lifetime*.
  - Normally, a new one is issued just prior to the expiration of the old one.

- In some cases, a certificate may need to be revoked prior to its expiration:
  - User's secret key is assumed to be compromised.
  - User is no longer certified by this CA.
  - CA certificate is assumed to be compromised.

X.509: Certificate Revocation List (CRL)

- Each CA maintains a list of all revoked not-expired certificates.
  - issued by that CA to users
  - issued to other CAs

- Certificate Revocation List (CRL) posted to the directory is signed by the issues and includes:
  - issuer's name
  - list creation date
  - next CRL creation date
  - revoked certificate entries (serial number and revocation date)
Two basic Certificate Revocation List delivery models:

- **Polling**: the current CRL is requested by the certificate user when he/she needs key on a digital certificate
  - Problem: time delay between revocation and publication
- **Pushing**: the new CRL is delivered by the CA to the user as soon as new revocation occurs
  - Problems: storage of new pushed CRLs even if irrelevant and danger of interception and deletion
X.509: Authentication Procedures

- Three alternative authentication procedures
  - Each use public-key signatures
  - Each assumes that two parties know each other's public key.
    - either obtained from Directory
    - or obtained in an initial message

X.509: One-way Authentication

- A single transfer of information from one user (A) to another (B) and establishes the following:
  - Identity of A and message generated by A
  - Message is intended for B
  - Integrity and originality of the message.
In addition, two-way authentication establishes the following:

- The identity of B and that the reply message is generated by B (the target of the first message).
- The message is intended for A.
- The integrity and originality of the reply.

---

Final message from A to B contains a signed copy of the nonce ($r_B$) received from B.

- Eliminates the need to check timestamps.
- Used when synchronized clocks are not available.
X.509 Version2 Inadequacies and Version3 Solution

Insufficient information conveyed in the certificate
- Subject field issues
  - inadequate to identify key owner
  - inadequate for many applications (that require, for example, e-mail or URL)
- No security policy information
- No method to limit damage (in case of faulty or malicious CA)
- No key differentiation
- Solution: two approaches
  - either add fields to version 2 format
  - or add optional extension fields (!)

X.509 Version 3 Certificate

Note: public key infrastructure in Windows 2000 supports X.509 version 3 certificates.

The definitions for the Version 3 fields are:
- **Version**: Version of the certificate format; for example, version 3 (code is 2).
X.509 Version 3 Certificate

- **Certificate Serial Number:** The unique integer that is assigned by the issuing CA.
  - The CA maintains an audit history for each certificate so that certificates can be traced by their serial numbers.
  - Revoked certificates also can be traced by their serial numbers (and the issuing CA’s name).

- **Certificate (Signature) Algorithm Identifier:** The public key cryptography and message digest algorithms that are used by the issuing CA to digitally sign the certificate.
- **Issuer Name:** The name of the issuing CA such as:
  - X.500 directory name
  - Internet e-mail address
  - X.400 e-mail address
  - URL
X.509 Version 3 Certificate

- **Validity Period**: The certificate’s start and expiration dates.
  - define the interval during which the certificate is valid, although the certificate can be revoked before the designated expiration date.

- **Subject**: The name of the subject (owner) of the certificate such as:
  - X.500 directory name
  - Internet e-mail address
  - URL

- **Subject Public-Key Information**: The public key and the public key cryptography algorithm.
  - The algorithms for which the public key set can be used, such as digital signing, secret key encryption, and authentication.
X.509 Version 3 Certificate

- **Issuer Unique Identifier:** Optional information (bit string) for uniquely identifying the issuer, when necessary.

- **Subject Unique Identifier:** Optional information (bit string) for uniquely identifying the subject, when necessary.

**Extensions:** Additional information that can be specified for optional use by public key infrastructures. Common extensions include a list of specific uses for certificates (for example, S/MIME secure mail or IPSec authentication), CA trust relationship and hierarchy information, a list of publication points for revocation lists, and a list of additional attributes for the issuer and subject.
X.509 Version 3 Certificate

- Certification Authority’s Digital Signature: The CA’s digital signature of all the previous fields, which is created as the last step in generating the certificate. (Called Encrypted)

- 3 extension categories
  - Key and policy information
  - Subject and issuer attributes
  - Certification path constraints
X.509 Extensions: Key and Policy

- Subject and issuer keys information
- Indicators of certificate policy
- Extension fields
  - Authority key identifier (to differentiate keys of the same CA)
  - Subject key identifier (to differentiate keys of the same subject)
  - Key usage (bit string for 9 possibilities, such as key and/or data encryption, signature verification on certificates/CRLs, …)
  - Private-key usage period (for signatures)
  - Certificate policies (used for issuing and for certificate usage)
  - Policy mappings (from CA to CA, for matching policies of different CAs)

X.509 Extensions: Certificate Subject Attributes

- Alternate names for either the certificate subject or the certificate issuer

- Extension fields
  - Subject alternative name (additional identities to be bound to the subject)
  - Issuer alternative name (to associate, e.g., internet style identities to issuer)
  - Subject directory attributes (such as DoB or clearance, to be used by X.500 directory)
X.509 Extensions: Certification Path Constraints

- Provide constraints for certificates issued by CAs for other CAs.

- Extension fields
  - Basic constraints (can subject be CA and length of allowed certification path from this CA)
  - Name constraints (name space for allowed subjects in subsequent certificates)
  - Policy constraints (for path validation, either prohibiting or requiring policy)

Vulnerability and Exploits

- In 2005, shown "how to use hash collisions to construct two X.509 certificates with identical signatures and different public keys", using a collision attack on the MD5 hash function.
- In 2008, presented a practical attack to create a rogue Certificate Authority, accepted by all common browsers, by exploiting the issuing X.509 certificates based on MD5.
- X.509 certificates based on SHA-1 appeared to be secure until April 2009 when researchers produced a method to increases the likelihood of a collision
- There are implementation errors with X.509 that allow e.g. falsified subject names using null-terminated strings or code injections attacks in certificates
- Implementations suffer from design flaws, bugs, different interpretations of standards and lack of interoperability.
  - Many implementations turn off revocation check and policies are not enforced