Authentication

• Digital signature validation proves:
  – message was not altered in transmission
  – came from owner of the private key

• How does a “relying party” know to whom the private key belongs?
  – Key Servers
  – Certificates
Key Server

“What is the public key of identity I?”

“The public key of identity I is K.”

• The key server stores [identity, public key] pairs
• The key request can be in plaintext
• The key server reply is encrypted using the private key of the key server
• The public key of key server is known to the relying party
• The key server can be a point of attack or performance bottleneck
• The key server must be trustworthy
• Observations:
  • the relying party only cares about the reply
  • the reply can be precomputed and distributed
Authentication using a Key Server
Needham-Schroeder Protocol
Certificates

- the certificate
  - contains an (identity, public key) pair
  - is signed with the private key of the CA

- the repository
  - need not be trusted
  - is read-only
  - may be duplicated for performance

- the certificate can be “pushed” to the relying party
Chain of Trust

trusted CA (root CA, trust anchor)

presented identity certificate

identity

signed by

signed by

signed by

identity
X.509 Certificate Format

- **VERSION**: v1 or v2 or v3
- **SERIAL NUMBER**: 12345
- **SIGNATURE ALGORITHM**: RSA with SHA-1
- **ISSUER**: C=US, S=VA, O=RSA Labs
- **VALIDITY**: 1/1/01 - 1/1/02
- **SUBJECT**: C=US, S=VA, O=RSA Labs, CN=Russell Housley
- **SUBJECT PUBLIC KEY INFO**: RSA, 48...321
- **ISSUER UNIQUE ID**: ACBDEFGH
- **SUBJECT UNIQUE ID**: RSTUVWXYZ
- **EXTENSIONS**: 
- **SIGNATURE**: 

Diagram:

- 12345 → VERSION
- C=US, S=VA, O=RSA Labs → ISSUER
- C=US, S=VA, O=RSA Labs, CN=Russell Housley → SUBJECT
- ACBDEFGH → ISSUER UNIQUE ID
- RSTUVWXYZ → SUBJECT UNIQUE ID
- Extensions
Example Certificate

Certificate:

Data:

Version: 3 (0x2)
Serial Number: 1097588 (0x10bf74)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=US, ST=Massachusetts, O=Massachusetts Institute of Technology, OU=Client CA v1
Validity
Not Before: Jul 31 14:07:49 2000 GMT
Not After: Jul 31 14:07:49 2001 GMT
Subject: C=US, ST=Massachusetts, O=Massachusetts Institute of Technology, OU=Client CA v1, CN=Jeffrey I Schiller/Email=jis@MIT.EDU
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:cf:01:0a:e5:f1:3c:60:c1:f2:c1:ca:99:96:1d:
e0:4a:6c:e8:abf4:cb:64:ea:c9:33:f7:efa9:8f:
8f:11:bf:1a:ef:7b:2d:06:ef
Exponent: 65537 (0x10001)

X509v3 extensions:
X509v3 Key Usage:

1.2.840.113554.1.3.1:
0:....]/e;ii;.....m......j....Nr....$wF..t...QZ...
Signature Algorithm: md5WithRSAEncryption
0a:58:1d:53:35:d0:75:82:9d:2a:e7:12:35:3f:88:60:cc:a3:
2a:6e
Revocation

- Is a certificate still valid?
  - Private key compromise
  - CA compromise
  - Affiliation changed
  - Superseded
  - CA ceased operation
  - ...

- Certificate Revocation List (CRL) provides a list of the unexpired certificates that should no longer be used
CRL Format

- v1 or v2
- C=US, S=VA, O=RSA Labs
- 12/2/01
- RSA with SHA-1
- 11/25/01
- 12345
- 9/27/01
- 10
## PKIX Elements

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>PRIMARY ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Entity</td>
<td>End Entity is a generic term used to denote end-users, devices (e.g., servers, routers), or any other entity that can be identified in the subject field of a public key certificate. End entities typically consume and/or support PKI-related services.</td>
</tr>
<tr>
<td>Certification Authority (CA)</td>
<td>The CA is the issuer of certificates and (usually) CRLs. It may also support a variety of administrative functions, although these are often delegated to one or more Registration Authorities.</td>
</tr>
<tr>
<td>Registration Authority (RA)</td>
<td>The RA is an optional component that can assume a number of administrative functions from the CA. The RA is often associated with the End Entity registration process, but can assist in a number of other areas as well.</td>
</tr>
<tr>
<td>Repository</td>
<td>A repository is a generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by End Entities.</td>
</tr>
<tr>
<td>CRL issuer</td>
<td>The CRL Issuer is an optional component that a CA can delegate to publish CRLs.</td>
</tr>
</tbody>
</table>
Role of the CA

- Verifies certificate request information
- Generates and digitally signs the certificate
- Revokes certificate if information changes
- Revokes certificate if private key is disclosed
- Support certificate hierarchies
- Optional services
  - Key generation
  - Issue hardware token
CA Topologies

Hierarchy

Mesh
Cross Certification