Public Key Infrastructure

R. E. Newman
CISE Department
UF
Gainesville, FL 32611−6120
nemo@cise.ufl.edu
+352.392.1488/1220 fax

The Key Management Scaling Problem

- Shared symmetric key
- Asymmetric key pair

With one symmetric key per pair, quadratic explosion
Central KDC can distribute session keys on demand (Kerberos, etc.)
Public Key approach only requires linear number of keys –
but still need reliable way to associate public key with entity...
Key Distribution Center (KDC)

Trusted third party
Should be able to authenticate requestors (at least implicitly)
Should be able to provide requestors with information that allows them
to authenticate themselves to others
Should provide keys to parties as warranted over secure channel
If KDC compromised, then all is lost

Certification Authority (CA)

Trusted third party
Should be able to provide requestors with information that allows them
to authenticate themselves to others
Signs certificates (that may hold keys, etc.)
Directory may use insecure channel for communication
CA need not be on-line
Compromised CA can permit spoofing, but cannot decrypt communications
established between the true parties
Heirarchical trust relationships, organizations
Scaling – for distribution, reliability, performance
Action at appropriate level (locality)
Cross–hierarchy trusts

X.509 Certificates

<table>
<thead>
<tr>
<th>Version</th>
<th>Cert. Serial #</th>
<th>default is 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>algorithm</td>
<td>unique (within CA) integer given to certificate</td>
</tr>
<tr>
<td></td>
<td>parameters</td>
<td>signature algorithm ID and parameters</td>
</tr>
<tr>
<td></td>
<td>Issuer Name</td>
<td>X.500 name of issuing CA</td>
</tr>
<tr>
<td></td>
<td>start</td>
<td>period of validity</td>
</tr>
<tr>
<td></td>
<td>end</td>
<td>user name associated with certificate (issuee)</td>
</tr>
<tr>
<td></td>
<td>Subject Name</td>
<td>subject public key information</td>
</tr>
<tr>
<td></td>
<td>algorithm</td>
<td>optional bit string field to identify uniquely</td>
</tr>
<tr>
<td></td>
<td>parameters</td>
<td>issuer in case of X.500 name reuse</td>
</tr>
<tr>
<td></td>
<td>key</td>
<td>optional bit string field to identify uniquely</td>
</tr>
<tr>
<td></td>
<td>Issuer</td>
<td>subject in case of X.500 name reuse</td>
</tr>
<tr>
<td></td>
<td>Unique ID</td>
<td>optional extensions</td>
</tr>
<tr>
<td></td>
<td>Subject</td>
<td>signature – hash of other fields encrypted</td>
</tr>
<tr>
<td></td>
<td>Unique ID</td>
<td>with issuing CA’s private key</td>
</tr>
<tr>
<td></td>
<td>Extensions</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>algorithms</td>
<td>Versions</td>
</tr>
<tr>
<td></td>
<td>parameters</td>
<td>Version 3</td>
</tr>
<tr>
<td></td>
<td>encrypted</td>
<td>Version 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version 1</td>
</tr>
</tbody>
</table>
To get B’s public key reliably, A must
Obtain copy of B’s certificate from a directory service
Build certification path (certificate chain) from root CA to certificate
Verify temporal validity of all certificates along certificate chain
Verify all signatures along certificate chain
Check all CRLs to ensure no certificate on chain was revoked
A has the root’s public key from the start

X.509 Certificate Binding Hierarchy

A→B: CA5<<A>>
A→B’: CA5<<A>>, CA1<<CA5>>, CA4<<CA1>>
A→B’’: CA5<<A>>, CA1<<CA5>>, CA0<<CA1>>, CA2<<CA0>>, CA6<<CA2>>
...or...
CA5<<A>>, CA2<<CA5>>, CA6<<CA2>>
certification path can use X.500 Directory Information Tree (DIT) or
DIT plus cross-registration short-cuts
X.509 Certificate Revocation

Each CA maintains a CRL (Certificate Revocation List)
May have ICRL (individual) and CACRL (CAs)
  vulnerability in specification – look the same
Must check the CRL for each CA on certification path

Certificate Revocation Lists

<table>
<thead>
<tr>
<th>algorithm parameters</th>
<th>signature algorithm identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer name</td>
<td></td>
</tr>
<tr>
<td>This update date</td>
<td>revoked certificates</td>
</tr>
<tr>
<td>Next update date</td>
<td></td>
</tr>
<tr>
<td>user cert. serial #</td>
<td></td>
</tr>
<tr>
<td>revocation date</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>algorithms parameters</th>
<th>signature signed</th>
</tr>
</thead>
</table>
X.509 Certificate Version 3 Extensions

- May need more info than subject field alone gives
- May need security policy information
- Need to limit damage from fault CA
- Need to be able to identify separate keys used by same subject (encrypt, verify)

Extension fields provide this in a flexible way

- Extension identifier
- Extension criticality – can this be ignored safely?
- Extension value

Three main categories –

- Key and policy info
- Subject and issuer attributes
- Certification path constraints

Certification Path Constraint Extensions

Constraints included in CA certificates for other CAs

May restrict types of certificates that can be issued by the subject CA, or that may occur later in the chain

Include:

- Basic constraints – indicates if subject can act as a CA; may constrain certification path length
- Name constraints – constrains name space of all subject names in rest of certification path
- Policy constraints – may specify explicit policy identification or may inhibit policy mapping for rest of path
Key and Policy Extensions

Certificate Policy = named set of rules that indicates applicability of a certificate to a particular community or class of application with common security requirements.

Includes:

- Authority key identifier – distinguishes among multiple signing keys for the CA
- Subject key identifier – distinguishes among multiple keys for subject (e.g., verification key, encryption key; useful for updating keys also)
- Key usage – restricts ways in which the key should be used:
  - digital signature, non-repudiation, key encryption, data encryption,
  - key agreement, CA signature verification on certificates, CA signature verification on CRLs
- Private key usage period – signing key usually shorter lived than verification key
- Certificate policies – list of policies the certificate supports, along with optional qualifying information
- Policy mappings – used only in certificates issued by CAs for other CAs; allows issuer to indicate equivalencies between policies of issuer and subject CA domains

Certificate Subject and Issuer Attribute Extensions

Support for alternative names, alternative formats
Can provide additional information about a subject for identification purposes (e.g., email address, postal address, position title, photograph, etc.)

Includes:

- Subject alternative name – one or more alternative names in a variety of forms; supports IPSEC, email, EDI, etc.
- Issuer alternative name – ditto for issuer
- Subject directory attributes – any X.500 directory attribute values for subject