Tutorial Title:  Usage of Certificate Policies in a PKI to Model Real-World Trust Relationships

Instructor:  Dr. Sarbari Gupta  
Director of Information Security  
CygnaCom Solutions  
7927 Jones Branch Drive, Suite 100W  
McLean, VA 22102  
(703)848-0883 (Voice)  
(703)848-0996 (FAX)  
gupta@cygnacom.com (Email)

Summary of Topics to be addressed in session:

  X.509 v3 certificates and certificate policy processing  
  Trust relationships in the business world  
  Modeling trust relationships using X.509 policy constructs  
  Policy Support Features in PKI Entities  
  Anomalies in X.509 policy handling and a proposed fix

Short Bio of Speaker:

  Dr. Sarbari Gupta has been active in the security field for over 12 years. As the Director of Information Security, she has oversight of the security consulting, development and integration teams at CygnaCom Solutions. She participates in a number of standards organizations. Her interests include PKI technology, key recovery, and secure applications.
Usage of Certificate Policies in a PKI to Model Real-World Trust Relationships

October, 1999

Dr. Sarbari Gupta
CygnaCom Solutions
7927 Jones Branch Drive, Suite 100W
McLean, VA 22102
http://www.cygnacom.com
sgupta@cygnacom.com
(703)848-0883
Outline

• X.509 Version 3 certificates and certificate policy processing
• Trust relationships in the business world
• Modeling trust relationships using X.509 policy constructs
• Policy support Features in PKI entities
• Anomalies in X.509 policy handling and a proposed fix
Public Key Certificate

A digital document that binds an entity (name, id) to a specific public key. A trusted third party (certification authority) establishes the binding using a digital signature.
Public Key Infrastructure (PKI)

A digital infrastructure that provides the needed levels of confidence to users of a public key that the associated private key is owned by the correct subject (person or system).

A PKI also provides a means of distributing public keys over an untrusted medium.
PKI Architectural Entities

**Certification Authority**
A trusted entity that:
- Verifies and vouches for the identity of subscribers
- Generates and signs Public Key Certificates
- Revokes Public Key Certificates
- Publishes Public Key Certificates and Certificate Revocation Lists in Directory Servers
- Operated under control of Security Officer(s)

**Repository**
Contains valid Public Key Certificates and Certificate Revocation Lists

**Subscriber**
A entity that:
- Generates asymmetric key pairs
- Requests public key certificates from CAs
- Receives issued certificates
- Uses private key in crypto operations

**Relying Party**
A entity that:
- Looks up peer certificates in Repository
- Validates peer certificates and certificate paths in order to establish trust in peer public key
- Uses peer public key in crypto operations
X.509 Version 3

Version 1:

• published in 1988 as part of the X.500 Directory recommendations,
• defines a standard, basic, certificate format.

Version 2:

• released when X.500 was revised in 1993,
• adds two more fields to support directory access control.

Version 3:

• adds extensions for performance, security, and limiting trust,
• adds support for cross-certification of CAs,
• adds X.500 naming constraints,
• adds support for management of policies
# X.509 v3 Certificate Profile

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version (v3)</td>
<td>Version Number</td>
</tr>
<tr>
<td>serial number</td>
<td>Unique Integer assigned by Certificate Issuer</td>
</tr>
<tr>
<td>signature algorithm id</td>
<td>Algorithm ID used to sign certificate</td>
</tr>
<tr>
<td>issuer name</td>
<td>X.500 DN of the certificate signing authority</td>
</tr>
<tr>
<td>validity period</td>
<td>Time period within which certificate is valid</td>
</tr>
<tr>
<td>subject name</td>
<td>X.500 DN of the subject entity</td>
</tr>
<tr>
<td>subject public key info</td>
<td>Algorithm ID and public key of the subject</td>
</tr>
<tr>
<td>issuer unique identifier</td>
<td>Additional identifying info to disambiguate issuer name</td>
</tr>
<tr>
<td>subject unique identifier</td>
<td>Additional identifying info to disambiguate subject name</td>
</tr>
<tr>
<td>extensions</td>
<td>Zero or more extension fields for the certificate</td>
</tr>
<tr>
<td>signature</td>
<td>Digital signature by the CA over the other certificate fields</td>
</tr>
</tbody>
</table>
X.509 v3 Certificate Standard Extensions

Authority Key Identifier  Subject Key Identifier
Key Usage               Extended Key Usage
Private Key Usage Period
Certificate Policies     Policy Mappings
Subject alternative name Issuer alternative name
Subject directory attributes Basic constraints
Name constraints     Policy constraints
CRL Distribution Points
X.509v3 Policy Extensions

- Certificate Policies
- Policy Mappings
- Policy Constraints
Certificate Policy

A certificate policy is defined as “a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements.” ¹

Certificate Policy

A CP addresses requirements for:

– Key generation
– Identity proofing
– Certificate and CRL generation and distribution
– Certificate Update, Renewal, and Re-key
– Certificate token initialization, programming, and management
– Privilege and Authorization Management
– System Management Functions (e.g. audit, certificate tracking, archiving)
– responsibilities and liabilities of the relevant parties
– security controls (technical, personnel, physical, and procedural)
Certificate Policy Extension

- Assertion of the policies under which the certificate was issued, indicating the purposes for which it may be used
- Multiple policies may be asserted
- **Policy qualifiers** allow further qualification of each policy asserted. IETF PKIX defines two qualifiers:
  - CPS Pointer
  - User Notice
Policy Mapping Extension

- Used in CA Certificates only
- Asserts equivalency relations between policies across disparate policy domains
Policy Constraints Extension

- Used in CA certificate only
- Asserts two types of policy related constraints
  - *Inhibit Policy Mapping* - indicates policy mapping is to be inhibited while processing subsequent certificates
  - *Require Explicit Policy* - indicates that subsequent certificates need to include an acceptable policy identifier
Policy Authority and Policy Domain

Policy Authority

An entity that defines (one or more) certificate policies, and assigns certificate policy identifiers.

Policy Domain

The set of policies administered by a single Policy Authority comprise a policy domain.
Certificate Path Processing

- Relying Party knows and trusts a CA’s Public Key; called trust anchor
- Relying Party has the Subscriber’s Public Key certificate
- Relying Party develops a chain of certificates beginning with a certificate signed by the trust anchor and ending with the Subscriber’s certificate
X.509 v3 Path Processing(I)

- **Inputs:**
  - initial-policy-set
  - initial-explicit-policy indicator
  - initial-policy-mapping-inhibit indicator

- **Outputs:**
  - success/failure of path validation
  - set of constrained policies and their qualifiers
X.509 v3 Path Processing(II)

• State Variables:
  – user-constrained-policy-set -- initialized to initial-policy-set
  – authority-constrained-policy-set -- initialized to any-policy
  – explicit-policy-indicator -- initialized to initial-explicit-policy indicator
  – policy-mapping-inhibit-indicator -- initialized to initial-policy-mapping-inhibit indicator
X.509 v3 Path Processing(III)

• **Return Failure if any of these checks fail:**
  – non-policy checks (e.g., signature verifies, dates are valid, certificate chains correctly, subject name in permitted namespace, etc.)
  – *if explicit-policy-indicator* is set
    • certificate policies extension contains at least one policy from the *user-constrained-policy-set*
  – *if certificate policies extension is critical*
    • update *authority-constrained-policy-set* by intersecting it with the certificate policies extension. Check that *authority-constrained-policy-set* is non-empty

• **Return success if this is the end certificate**
X.509 v3 Path Processing(IV)

- **Update the state variables:**
  - update non-policy state variables
  - if *explicit-policy-indicator* is not set
    - update *explicit-policy-indicator* based on the presence of the \texttt{requireExplicitPolicy} constraint in the certificate
  - if *policy-mapping-inhibit-indicator* is not set
    - if policy mapping extension is present
      - update *user-constrained-policy-set* based on mappings
      - update *authority-constrained-policy-set* based on mappings
    - update *policy-mapping-inhibit-indicator* based on the presence of the \texttt{inhibitPolicyMapping} constraint in the certificate
Trust/Authority Relationships in the Business World

• Intra-Organizational:
  – hierarchical (one or more levels)
  – networked
  – combination

• Inter-Organizational:
  – networked with no trust propagation
  – networked with trust propagation
Hierarchical Trust Relationships
Networked Trust Relationships

Sub-organizational unit

Direction of trust flow
Combination Trust Relationships

Sub-organizational unit

Direction of trust flow

Diagram showing the combination of trust relationships with sub-organizational units and the direction of trust flow.
Networked with no Trust Propagation
Networked with Trust Propagation
Modeling Trust Relationships using X.509 constructs

• Assumptions
  – an organization has a single policy authority
  – an organization comprises a single policy domain
  – a sub-organizational unit may operate its own Certification Authority
Business Goals for Trust/Authority Modeling

- superior CAs can apply different policies than subordinate CAs
- superior CAs can restrict the policies asserted by subordinate CAs
- new policies can be added dynamically to a existing policy domain without need to reissue superior certificates
Hierarchical Policy Domain (I)

- Single rooted hierarchy
- CAs closer to root apply higher assurance policies than CAs farther from the root
- Superior CAs assert a superset of the policies asserted by subordinate CAs

H = High Assurance
M = Medium Assurance
L = Low Assurance

CA X using policy M
Certificate asserting policy H
Policy Domain

Subscriber
Hierarchical Policy Domain (II)

- Single rooted hierarchy
- CAs closer to root apply higher assurance policies than CAs farther from the root
- superior CAs assert their local policy independent of the policies asserted by subordinate CAs
- new lower assurance policies may be added to the PKI dynamically

H = High Assurance
M = Medium Assurance
L = Low Assurance
VL = Very Low Assurance

CA X using policy M
Certificate asserting policy H
Policy Domain
Networked Policy Domain

- no common root
- relying party trusts local CA
- CAs may establish pair-wise trust relations

A = secure mail
B = secure web connections
C = secure network logon

CA X using policy M
Certificate asserting policy H
Policy Domain
Trust Propagation Across Policy Domains

- Cross-certifying CAs assert all possible policy equivalencies between their respective domains
- Relying party can authenticate Subscriber 2 as well as Subscriber 1 using certificate path via cross-certificate

A = secure mail
B = secure web connections
C = secure network logon

Relying Party (accepts H,M,L)
Subscriber 1
Subscriber 2

CA-X → CA-N
CP = M
PM = {(M, B), (L, C)}
Features for Maximal Support of Policy Processing

Affected Entities in the Public Key Infrastructure:

• Certification Authority
• Subscriber
• Relying Party
Certification Authority Features

Support for:

- multiple certificate policies in issued certificates
- selectable set of certificate policies for each certificate
- multiple policy mappings in cross-certificates
- inclusion of policy mappings independent of policies asserted
- self-signed certificates for use as trust anchors
Subscriber Features

Support for:
- multiple certificate policies in certificate requests
- self-signed certificates for use as trust anchors
Relying Party Features

Support for:
• full X.509 path processing algorithm
• use of a set of policies as initial-policy-set
• configurable contents of initial-policy-set
• processing of policy qualifiers
• use of self-signed certificates as trust anchors
Anomalies in X.509 Policy Processing

• Three anomaly scenarios
• Identification of underlying flaws in X.509 policy handling
• Proposed Fix to policy anomalies
Scenario 1

It appears to be difficult for a CA issuing a cross-certificate to a subject CA to restrict the policies that may be asserted by the subject CA.

If \textit{initial-policy-set} is \{A, B\}, then the above chains will be valid. It is difficult for \textit{CA-X} to restrict \textit{CA-N} to only asserting policy \textit{b} in \textit{Cert2} and \textit{Cert2’}
Scenario 2

There appears to be no way for a superior CA to restrict the policies that may be asserted by a subordinate CA. Assume CA-X would like to restrict CA-N to only asserting policy B.

If initial-policy-set is \{A, B\}, then the above chain will be valid. It is difficult for CA-X to restrict CA-N to only asserting policy B in Cert2.
Scenario 3

It appears to be difficult for a superior CA (\textit{CA-X}) to restrict a subordinate CA (\textit{CA-N}) from asserting policy \textit{High}.

If initial-policy-set is \{High, Med\}, then the above chain will be valid.
Underlying Policy Flaws in X.509

All 3 anomaly scenarios are symptomatic of two underlying flaws:

1) The certificatePolicies extension is overloaded to signify:
   – policies under which certificate was issued, indicating the purposes for which it may be used
   – polices that may be asserted by subordinate CAs, through use of the “critical” flag

2) The path processing logic does not check policies asserted in a certificate against authority-constrained-policy-set
Fix to Policy Flaws in X.509 (I)

• Restrict the usage of the certificatePolicies extension to signify:
  – policies under which certificate was issued
  – criticality flag has no effect on path processing

• Add a new extension, permittedPolicies to signify:
  – policies that may be asserted by subordinate CAs
Fix to Policy Flaws in X.509 (II)

• Update path processing logic such that the `explicit-policy-indicator` state variable is not used

• Update path processing logic such that the policies asserted in a certificate are always checked against the constraints placed by the user as well as the authorities.
New X.509v3 Extensions

• *permittedPolicies*: populated in a CA certificate to restrict the set of policies that may be asserted by subordinate CAs
Updates to Path Processing (I)

**Inputs**: Delete an input
- `initial-explicit-policy-indicator`

**Outputs**:
- if validation was successful, the set of acceptable policies and the corresponding qualifiers contained in the end certificate.

**State variables**: Delete a variable
- `explicit-policy-indicator`

**Initialization Step**: No change

**Local Variables**: Add a local variable
- `acceptable-policies` - policy identifiers asserted in the current certificate that are considered acceptable to the certificate user, as well as the preceding authorities
Updates to Path Processing (II)

• Processing of all certificates:
  c) If $user\text{-constrained-policy-set}$ is not $any-policy$, compute the intersection of the $user\text{-constrained-policy-set}$, the $authority\text{-constrained-policy-set}$, and the identifiers within the certificate policies field and put the result as the $acceptable\text{-policies set}$. Check that the $acceptable\text{-policies set}$ is not the NULL set.
  [UPDATE TO EXISTING STEP]

If all checks pass, and this is the end certificate, return the policies in the $acceptable\text{-policies set}$ and the corresponding policy qualifiers that appear in the current certificate.
Updates to Path Processing (III)

• Processing of intermediate certificates:
  
  d) Delete update of *explicit-policy-indicator* state variable
  
  e) If *policy-mapping-inhibit-indicator* is not set: [NO CHANGE]
     – process any policy mapping extension with respect to … *user-constrained-policy-set* and add appropriate policy identifiers …
     – process any policy mapping extension with respect to … *authority-constrained-policy-set* and add appropriate policy identifiers …
  
  f) If the *permittedPolicies* extension is present, compute the intersection of the policies in that extension and the *authority-constrained-policy-set* and put the result as the new value of the *authority-constrained-policy-set* [NEW STEP ADDED]
Advantages of Proposed Fix

- Semantics of all existing extensions remain unchanged
- No restrictions on the way policies may be deployed within policy domains, when new extension is not used
- Minimal changes (through augmentation rather than replacement) in path processing algorithm
- Full backward compatibility with existing CAs and issued certificates
- Ability of superior CAs to restrict policies that may be asserted by subordinate CAs
Usage of certificatePolicies Extension

• This extension should include policy identifiers only for the policies that were used in issuing the subject certificate. For example, in Scenario 1, CA V should assert only \{H\}, instead of \{H,M,L\}

• The semantic description of this extension should be changed to “… policy information terms indicate the policy under which the certificate has been issued indicating the suitability of the certificate for specific purposes and applications”
Usage of permittedPolicies Extension

• When dynamic addition of lower assurance policies is desirable within a policy domain, the permittedPolicies extension should not be used within CA certificates within the domain.

• When the assertion of policies by subordinate CAs within a domain is to be restricted, the permittedPolicies extension should be populated with ALL policies that may be asserted be subordinate CAs.
Usage of PolicyMappings Extension

- This extension should be used to convey policy equivalence relations between two policy domains. Thus, when cross-certifying between policy domains, the policyMappings extension may include all possible equivalency statements between policies in the subject domain and issuer domain. This extension is not required to be limited to equivalency relations corresponding to the policies asserted within the cross-certificate.