Design and Implementation of LDAP Component Matching for Flexible and Secure Certificate Access in PKI

Sang Seok Lim  
IBM Research  
slim@us.ibm.com

Jong Hyuk Choi  
IBM Research  
jongchoi@us.ibm.com

Kurt Zeilenga  
IBM Linux Technology Center  
zeilenga@us.ibm.com

4th PKI R&D Workshop, NIST, Gaithersburg MD 2005
Outline

- Introduction
- LDAP-PKI Interoperability Limitation
- Interoperability Enabling Technologies
  - Component Matching
  - ASN.1 Awareness
  - Generic String Encoding Rule (GSER)
- OpenLDAP Implementation
  - ASN.1 Generation of ASN.1 Decoders / Matching Rules
  - Component Matching Architecture and Data Structures
  - Component Matching Optimizations
- Performance Evaluation
- Summary
OpenLDAP Component Matching: Overview

Enhanced PKI Certificate/CRL Repository

OpenLDAP Component Matching Improves PKI-LDAP Interoperation
- Simple DIT structuring
- No need to de-aggregate (or shred) PKI attributes
- High performance Certificate Repository
- First implementation of Component Matching in a pure LDAP server

Public Key Infrastructure
- Certificate Authority
- Registration Authority
- End Entity
- LDAP Repository

LDAP-PKI Interoperability Mismatch
- DIT Complexity
- Data Integrity Checking
- Performance Degradation

PKI CA
PKI User
Directory Service

Enhanced PKI Support
(Flexibility, Manageability, Security)

ASN.1 Awareness
Component Matching

LDAP Repository
Usage of LDAP in PKI

- **Certificate access protocols in PKI**
  - LDAP, HTTP, FTP, etc.

- **LDAP: a predominant way of implementing a PKI repository**
  - X.509 was originally specified in the X.500 context
  - LDAP is predominantly used directory protocol for the Internet
    - Various protocol operations: search, modify, update, etc
    - Authentication: Simple Authentication Security Layer (SASL)
    - LDAP Control
    - Access control

- **LDAP has X.500 DAP simplified by mainly**
  - String-based encoding
    - Incapable of preserving structural information
    - DAP uses ASN.1 encodings (BER/DER)
  - Direct TCP/IP mapping
  - Simple-protocol encoding
Certificate Access using LDAP

LDAP Search Filter

Find the certificate whose subject is
“cn=John,o=IBM,c=US”
and
keyUsage is
“non-repudiation”

End-entity (Client)

LDAP Request

LDAP Response

LDAP Certificate Repository

Certificate for non-repudiation

Certificate for non-repudiation for encryption

4th PKI R&D Workshop, NIST, Gaithersburg MD, April 19th 2005
Certificate Structure and Encodings (bits-on-the-line)

Pictorial Representation of X.509 Certificate ASN.1 Specification

The structural information of ASN.1 types specification is preserved in the encodings

DER: Distinguished Encoding Rules
LDAP Client Side Requirement

Find the certificate whose subject is “cn=John, o=IBM, c=US” and keyUsage is “non-repudiation”.

LDAP Request

userCertificate Attribute

version
serialNumber
signature
issuer
validity
subjectPublicKeyInfo
issuerUniqueIdentifier
subjectUniqueIdentifier
extensions
non-repudiation
Requirement: **Component-level** expressive power of LDAP search filters

Deficiency: LDAP’s incapability of composing a **component-level** filter!!

New definitions of component assertion and filter by Component Matching (RFC3687)

End-entity (Client)
LDAP Server Side Requirement
Requirement: Restore **structural information** of ASN.1 specifications for component-level matching

Deficiency: LDAP string encoding which do not preserve structural information

**ASN.1 awareness in LDAP directory server by using an ASN.1 compiler**
Certificate Syntax Specific Parsing

- Parsing the blobs (DER) by certificate-syntax specific decoders

- Limited and inflexible component-level filter composing
  - Example search filter

(userCertificate:certificateExactMatch=cn=CA,o=IBM,c=US$12345)

- Disadvantages
  - Inflexibility
- **Example search filter (use an attribute-level filter)**
  \[(&((x509KeyUsage=010000000)(x509Subject=cn=John,o=IBM,c=US)))\]

- **Certificate Parsing Server (XPS): automates the extraction process**

- **Disadvantages**
  - High storage requirements
  - Security Issue
    - Data integrity checking: matching is performed unsigned extracted attributes
  - Poor manageability
    - DIT Restructuring to support multiple certificate for an user
Directory Information Tree of Attribute Extraction

- DIT Restructuring for storing multiple certificates for a user

(a) DIT

(b) DIT of Attribute Extraction

4th PKI R&D Workshop, NIST, Gaithersburg MD, April 19th 2005
Example Component Matching Filter

User request statement:
“Find a certificate whose subject is cn=John,o=IBM,c=US and keyUsage is non-repudiation"
ASN.1 Awareness and Component Matching

- **ASN.1 awareness of LDAP directory servers**
  - Ability to utilize the structural information of an ASN.1 type.
  - Understand how to construct complex types by combining basic types and composite types of ASN.1

- **Component matching for ASN.1 awareness**
  - Define how to describe component filter and assertion for components
  - Define how to reference a component: *component reference*
  - Define a *generic way of matching* among components at an ASN.1 level
  - Generic String Encoding Rule (GSER)
    - Introduce structure information into UTF-8 based string encodings
    - The value of component assertion is encoded in GSER
Generic String Encoding Rule

- Define **UTF8 string encodings rule** for basic ASN.1 types and composite ASN.1 types
  - INTEGER, BOOLEAN, OCTET STRING etc
  - SEQUENCE, SET, CHOICE, etc

```plaintext
toBeSigned :: = SEQUENCE {
    version [0] EXPLICIT Version DEFAULT v1,
    serialNumber CertificateSerialNumber,
    signature AlgorithmIdentifier,
    issuer Name,
    validity Validity,
    subject Name,
    subjectPublicKeyInfo subjectPublicKeyInfo,
    ... extensions [3] EXPLICIT Extensions OPTIONAL
}
```

Example GSER Encodings of Certificate

```
{ version 2, 
  serialNumber 12345 , 
  signature { algorithm 1.2.840.113549.1.14, parameters NULL},
  issuer {{type cn, value IBM trust} , {type o, value IBM},{type c, value US}},
  validity {notBefore {2004 01 13 18 59}, notAfter {2005 01 13 18 59} },
  ... 
}
```
Our Framework of Component Matching Enabled OpenLDAP
An eSNACC Compiler and a GSER Back-end

- **What does the compiler do?**
  - Compile ASN.1 (Abstract Syntax Notation One) modules into
    - ASN.1 equivalent C data structures
    - Routines to convert to/from the internal (C or C++) representation from/to the corresponding BER/DER formats

- **GSER back-end**
  - Generate GSER encoding/decoding functions of given ASN.1 types

---

**Certificate ASN.1 Type**

```
toBeSigned ::= SEQUENCE {
  version [0] EXPLICIT Version DEFAULT v1,
  serialNumber CertificateSerialNumber,
  signature AlgorithmIdentifier,
  issuer Name,
  validity Validity,
  subject Name,
  subjectPublic_keyInfo subjectPublic_keyInfo,
  ... 
  extensions [3] EXPLICIT Extensions OPTIONAL
}
```

---

**GSER Encodings**

```
{ version v1, serialNumber 12345,signature {...},
  "issuer cn=CA,o=ibm ...
}
```

---

**C internal data structure**

```
typedef struct toBeSigned{
  AsnInt version;
  AsnInt serialNumber;
  struct AlgorithmIdentifier* signature;
  struct Name* issuer;
  ... 
} C internal data structure
```
Component Representation

- **Component**
  - The arbitrary part of attribute value that can be referenced or identified by Component Reference

- **Component representation**
  - Preserve ASN.1 structure information in the internal representation
  - Two parts
    - Data value
      - Value in a C internal data structure
    - Component descriptor for its value processing
      - Value-specific encoder/decoder/matching rule/extract
  - Component tree
    - All components are comprised of one tree

- **Component extraction**
  - Extract referenced components from a component tree
  - Generated by the eSNACC ASN.1 compiler automatically
Component Matching Implementation

- **Component assertion (GSER encoded)**
  - An assertion about the presence, or values of, components within an ASN.1 value
  - Component reference
    - Identifying the component part of a ASN.1 value.
  - Matching rule
  - Component assertion value in GSER

- **Component filter (GSER encoded)**
  - An “and”, “or”, “not” expression of ComponentAssertion, evaluates to either TRUE, FALSE or Undefined

- **Component equality matching rule**
  - allComponentsMatch and refined matching rules
  - Generated by the eSNACC ASN.1 compiler automatically
User request statement:
“Find a certificate whose subject is \textit{cn=John,o=IBM,c=US} and keyUsage is \textit{non-repudiation}”

New Matching Rule

\begin{verbatim}
(userCertificate:componentFilterMatch
 := and:
  item:{
    component “toBeSigned.subject”,
    rule distinguishedNameMatch,
    value “cn=John,o=IBM,c=US”
  },
  item:{
    component “toBeSigned.extension.*.extnValue.(2.5.29.15)”,
    rule bitStringMatch,
    value ‘010000000’B
  }

Component Assertion Value: BIT STRING in GSER
\end{verbatim}

Example Component Filter
Overall Operational Steps in Our Framework
Component Matching Optimization

- **Attribute/matching rule aliasing**
  - Backward compatibility for legacy clients
  - Avoid expensive extensible filter processing

```
x509certificateSerial=12345678

userCertificate:componentFilterMatch =:
  item:{
    component "toBeSigned.serialNumber",
    rule IntegerMatch,
    value 12345678
  }
```

<table>
<thead>
<tr>
<th>Alias Attribute</th>
<th>Aliased Attribute</th>
<th>Component Reference</th>
<th>Matching Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>x509certificateSerial</td>
<td>userCertificate</td>
<td>toBeSigned.serial</td>
<td>integerMatch</td>
</tr>
<tr>
<td>x509certificateIssuer</td>
<td>userCertificate</td>
<td>toBeSigned.issuer</td>
<td>distinguishedNameMatch</td>
</tr>
</tbody>
</table>

Example Attribute Aliasing Table
Component Matching Optimization Contd.

- **Component indexing**
  - Boost search performance by supporting component-level indexing
    - Indices on serial number, issuer name, version, etc

- **Component caching**
  - Eliminate certificate (DER) decoding overheads
    - 72usec/certificate : Intel Xeon 2.8GHz
  - Cache decoded internal representations of a certificate
Performance Evaluation

- **Directory population**
  - OpenSSL and Component Matching
    - DirectoryMark generated entries: 100k and 500k
  - Attribute extraction
    - Used XPS (OpenLDAP patch) to extract attributes

- **System under test (SUT)**
  - IBM xSeries 445 servers: 4-way Intel Xeon 2.8GHz with 12GB memory
  - Network connection: 1 Gbps Ethernet
  - Server: OpenLDAP 2.2.2 and Berkeley DB 4.3
  - Clients: DirectoryMark scripts (8-way IBM xSeries 445 servers)

- **Performance measurement**
  - Search throughput (operations/sec)
    - Matching against `serialNumber`
  - 100K entry add performance
Performance (Population / Search)

Directory Population Performance

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Component Matching</th>
<th>OpenSSL</th>
<th>Attribute Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>100K entries</td>
<td>178</td>
<td>167</td>
<td>815</td>
</tr>
<tr>
<td>DB Size(Mbytes)</td>
<td>234</td>
<td>234</td>
<td>410</td>
</tr>
</tbody>
</table>

Directory Search Performance

100k entries, DB cache = 1GBytes

500k entries, DB cache = 200MBbytes

4th PKI R&D Workshop, NIST, Gaithersburg MD, April 19th 2005
Summary

- Enhance LDAP-PKI interoperation for **flexibility** and **manageability**

- **Component matching enabled OpenLDAP server**
  - ASN.1 awareness
    - Supports any ASN.1 type from its specification
    - Supports GSER in attribute / assertion values
  - Component Matching
    - Supports searching of any Certificate attributes
  - First implementation of component matching in a pure LDAP server

- **Component optimization techniques**
  - Component aliasing, component indexing, and component caching
  - Enable high-performance, scalable LDAP certificate repository

- **Component matching enabled OpenLDAP** will help to use PKC more flexibly in many PKI applications of Internet2, OASIS, GRID…