

LDAP for the Java™ Net URL Framework

Part II: How to fetch CRLs from distribution points

Dieter.Bratko@iaik.tugraz.at

Stiftung Secure Information and Communication Technologies
IAIK, Graz University of Technology
August 2006, Copyright © by SIC/IAIK

Introduction

Unless the growing popularity of the Online Certificate Status Protocol (OCSP), certificate revocation lists (CRLs) are still most commonly used for providing revocation information about X.509 certificates. CRLs are publicly available from distribution points like HTTP or LDAP servers. A certificate usually contains a `CRLDistributionPoints` extension with a link to the location from where the corresponding `crl` can be obtained. You might think that is simple and straightforward to follow the link and download a `crl` from its distribution point. However, a `CRLDistributionPoints` extension may be structured in different ways making it already difficult to filter the information from where to get the revocation list.

This article shows how you can let IAIK-JCE do all the basic work for you to easily download a certificate revocation list from its distribution point. We first give a brief description of the `CRLDistributionPoints` certificate extension. Then we provide an example showing how to use IAIK-JCE for downloading a `crl` based on the information contained in the `CRLDistributionPoints` extension.

The `CRLDistributionPoints` extension

This chapter provides a short description of the `CRLDistributionPoints` extension. It should give you a feeling of the several possibilities how revocation information maybe linked. You may skip this section; understanding of the `CRLDistributionPoints` extension structuring is not absolutely required for using IAIK-JCE to download a `crl` from its distribution point.

The X.509 PKI and CRL profile defines the `CRLDistributionPoints` extension as ASN.1 SEQUENCE of `DistributionPoint` objects (see RFC 3280), each of which pointing to a location from where a CRL can be obtained:

```
CRLDistributionPoints ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint
```

```
DistributionPoint ::= SEQUENCE {  
    distributionPoint      [0]      DistributionPointName OPTIONAL,  
    reasons                [1]      ReasonFlags OPTIONAL,  
    cRLIssuer              [2]      GeneralNames OPTIONAL }
```

```
DistributionPointName ::= CHOICE {  
    fullName               [0]      GeneralNames,  
    nameRelativeToCRLIssuer [1]     RelativeDistinguishedName }
```

```
ReasonFlags ::= BIT STRING {  
    unused                 (0),  
    keyCompromise         (1),  
    cACompromise          (2),  
    affiliationChanged    (3),  
    superseded            (4),  
    cessationOfOperation  (5),  
    certificateHold       (6),  
    privilegeWithdrawn    (7),  
    aACompromise          (8) }
```

The `DistributionPointName` field of each `DistributionPoint` may be a `fullName` of type `GeneralNames` or an RDN relative to the `crl` issuer distinguished name (DN). In the first case the `fullName` field may represent a URI that points to the location from which to get the CRL. In the second case – if `nameRelativeToCRLIssuer` is set – the specified RDN has to be appended to the DN of the `crl` issuer. The DN of the `crl` issuer may be given in the `cRLIssuer` field of the distribution point, or – if `cRLIssuer` is not present – may be the DN of the certificate issuer. The `cRLIssuer` field only must be present if the corresponding `crl` is an indirect `crl` where the issuer of the `crl` is not the same as the issuer of the certificate for which revocation information shall be obtained.

The scope of each `crl` can be limited to some of the reasons given in the optional `ReasonFlags` component.

Downloading a CRL from a distribution point

Fortunately most commonly CRL distribution points refer to a (HTTP or LDAP) URL. Since LDAP by default is not supported by the `java.net` URL implementation – see Part 1 of this two-part article series (http://jce.iaik.tugraz.at/sic/support/technical_articles) – you first will have to register the IAIK-JCE LDAP protocol handler if you want to be able to get CRLs from LDAP distribution points:

```
System.getProperties().put("java.protocol.handler.pkgs",  
                           "iaik.x509.net");
```

To get a `CRLDistributionPoints` extension from an `X509Certificate` object, call method `getExtension` with the `OID` of the `CRLDistributionPoints` extension:

```
X509Certificate cert = ...;  
CRLDistributionPoints cRLDPs =  
    (CRLDistributionPoints)cert.getExtension(CRLDistributionPoints.oid);
```

Since more than only one `DistributionPoint` may be included, you must get an `Enumeration` of the `DistributionPoint` elements contained in the `CRLDistributionPoints` extension:

```
Enumeration e = cRLDistributionPoints.getDistributionPoints();
```

Now step through the enumeration and call method `loadCrl` of each `DistributionPoint` object to download the `crl` from the location the `dp` points to:

```
while (e.hasMoreElements()) {  
    DistributionPoint dp = (DistributionPoint)e.nextElement();  
    // download crl  
    X509CRL crl = dp.loadCrl();  
}
```

If you want to be sure that a particular distribution point actually refers to a `URI` you may call method `containsUriDpName` before downloading a `crl`.

Summing up the following source code fragment will download all `CRLs` from the distribution points of a `CRLDistributionPoints` extension that refer to a `URL`:

```
// register IAIK-JCE LDAP protocol handler  
System.getProperties().put("java.protocol.handler.pkgs",  
                           "iaik.x509.net");  
  
// get CRLDistributionPoints extension from a certificate  
X509Certificate cert = ...;  
CRLDistributionPoints cRLDPs =  
    (CRLDistributionPoints)cert.getExtension(CRLDistributionPoints.oid);
```

```
// get and step through all distribution points
Enumeration e = CRLDistributionPoints.getDistributionPoints();
while (e.hasMoreElements()) {
    DistributionPoint dp = (DistributionPoint)e.nextElement();
    If (dp.containsUriDpName()) {
        // download crl
        X509CRL crl = dp.loadCrl();
    }
}
```

If you have to deal with a distribution point that does not refer to a (HTTP or LDAP) URL, but uses the `nameRelativeToCRLIssuer` choice described in chapter 2, you may have to know the LDAP server URL (and maybe `crl/certificate issuer DN`) in advance. In this case use method `loadCrl(String ldapUrl, Name crlIssuer)` to download the revocation list (see IAIK-JCE Javadoc for more information). However, usually this is not required since most CAs use the `fullName` URL option to point to the location from where to get the CRL.

Summary

This article shows how to use IAIK-JCE to download certificate revocation lists from their distribution points without detailed knowledge about distribution point structuring and LDAP URL handling.

References

1. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile: <http://www.ietf.org/rfc/rfc3280.txt>
2. The LDAP URL format: <http://www.ietf.org/rfc/rfc2255.txt>
3. Java Naming And Directory Interface (JNDI): <http://java.sun.com/products/jndi/>
4. IAIK-JCE Toolkit: http://jce.iaik.tugraz.at/products/core_crypto_toolkits/jca_jce
5. Source Code Example: [GetCrlFromDp.java](#)