Network Working Group
Internet-Draft
Expires: August 23, 2003

L. Daigle
VeriSign, Inc.

February 22, 2003

IRIS Certificate and Key Registry
draft-daigle-iris-credreg-00.txt

Status of this Memo

Copyright Notice

Abstract

This document defines a credentials registry (credreg) and provides a specification in terms of an IRIS (draft-ietf-crisp-iris-core) registry schema. This registry enables location of certificates and public keys -- credential metadata can be searched to yield (pointers to) credentials. The schema extends the necessary query and result operations of IRIS to provide the functional information service needs for syntaxes and results used by credential registry administrators.

Table of Contents

1. Introduction

This document defines a credential (certificate and public key) registry (credreg), and describes an IRIS registry schema for the service using an XML Schema [4] derived from and using the IRIS [5] schema.

The schema given is this document is specified using the Extensible Markup Language (XML) 1.0 as described in XML [1], XML Schema notation as described in XML_SD [3] and XML_SS [4], and XML Namespaces as described in XML_NS [2].

It is important to note that XML is case sensitive. XML specifications and examples provided in this document MUST be interpreted in the exact character case presented to develop a conforming implementation.

2. Document Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [11].

3. Credentials Registry

The intent of the IRIS-based credentials registry is to provide a simple, public system to facilitate the location of certificates and public keys for particular users, roles and applications. This is not a replacement for trust service protocols such as XKMS [12], and is independent of any particular public key infrastructure. Results may include certificates and public keys directly, or pointers to access them using trust service protocols.

3.1 Data Model

The descriptive data included in the registry definition is based on the information a client is expected to have (in a query) and require (in a response) in order to locate actual credential information. The registry record is not a replacement for (or a super/subset of) a certificate format.

While the rest of this document describes various parts of the data model used, the IRIS-based XML schema specified in Section 4 is definitive for acceptable descriptive information and result record formats.

3.2 Interaction Model

The basic interaction is for the client to provide descriptive information about the entity (individual or application) for which the credential is sought, and the credreg server returns 0, one or more records that match the descriptive information. Those records contain the credentials, or pointers to them.

While the primary motive for this registry is to provide access to public credentials, it does inherit the basic access control mechanisms of IRIS.

3.2.1 Location of User Certificates and Keys

Applications such as encrypted e-mail require that an initiator have the (public) credentials of the recipient. If the initiator has not previously interacted with the recipient, their client software may not have a copy of the necessary credential. It is expected that the initiator would have the e-mail address and/or name of the intended recipient. The credentials registry provides a mechanism for using the information the initiator does have to locate the information they are not likely to have (the credential).

3.2.2 Location of Application Keys

As indirection becomes more common in application server and service provision, it becomes important to have a mechanism to look up the credentials of any given application. It has been suggested that a new DNS RR could be defined to hold such information. The credreg approach is to provide one single server (the credreg server) which holds the information necessary to lookup and retrieve application credentials.

3.3 Data Currency

The purpose of the registry is to provide information on the location of current credentials for individuals or applications. There is no provision to provide historical data or pointers.

3.4 Registry Server Independence

This document defines the behaviour of individual credentials registries. While there is no specific requirement that this be so, the driving model at the basis the specification is that each server will act as a registry for users and application servers in one or more (DNS) domains. Section 7 describes a service location convention that provides clients with a strategy for finding authoritative credreg server(s) for a given domain.

4. Schema Description

IRIS requires the derivation of both query and result elements by a registry schemas. These descriptions follow.

References to XML elements with no namespace qualifier are from the schema defined in Section 5. References to elements with the "iris" XML namespace qualifier are from the schema defined in IRIS [5].

4.1 IRIS Result Derivatives

4.1.1 <individualCredential>

The <individualCredential> result type must contain

- <individualCredentialHandle> -- a server-specified identifier for this specific credential
- <credentialApplication> -- the specific application for which the credential is intended (e.g., e-mail, instant messaging, etc). A special value of "any" is used if there is no particular application.
- one of
  - <publicKey> -- public key
  - <publicCert> -- public part of certificate
  - <credentialURI> -- a URI of any type referring to the credential material

and may contain any or all of

- <Name> -- personal name of the individual.
- <Role> -- individual's role. In some cases, the credential may be associated with a role (e.g., "administrator") rather than a specific person.
- <individualID> -- the ID associated with the individual. This is application-specific (e.g., an e-mail address for e-mail credentials).
- <domainName> -- the domain name of association for the credential.

4.1.2 <serverCredential>

The <serverCredential> result must contain

- <serverCredentialHandle> -- a server-specified identifier for this particular credential
- <serverName> -- the name of the server itself (as a domain or host name)
- <credentialApplication> -- an identifier for the application in question (e.g., "www"). Several applications may be operated on the same server, and it is necessary to be able to distinguish between them in order to provide the right credentials.
- one of
  - <publicKey>
  - <publicCert>
  - <credentialURI> -- any URI referencing the required credential material.

and may contain any or all of

- <domainName> -- the domain name of operation of the server
- <organizationName> -- the name of the organization responsible for the server.

4.1.3 <entityRefResult>

The <entityRefResult> contains an IRIS entity reference that can be used to retreive the credential result in the server. The client may request that results be returned as a set of <entityRefResult> so that a more compact reply is returned (though further dereferencing will be needed to get the credential material itself).

4.2 IRIS Query Derivatives

4.2.1 <findIndividualByID> query

<findIndividualByID> finds an individual's credentials by the ID associated with the credential. For example, this might be an e-mail or instant messaging (IM) address. The search may be constrained by specifying the identifier of the application for which the credential

is used. E-mail and IM addresses may be the same string; the application qualifier can be used to distinguish them. The client may request that the results be returned as entity references only (<entityRefsOnly>).

The <findIndividualByID&gt query returns a result set of <individualCredential> or of <entityRefResult> if the client requested entity references only. These are defined in Section 4.1.

Query fragment:

```
  <findIndividualByID>
    <exactMatch>leslie@thinkingcat.com</exactMatch>
    <credentialApplication>e-mail</credentialApplication>
  </findIndividualByID>
```

Response fragment:

```
  <individualCredential>
    <individualCredentialHandle>
         leslie.thinkingcat.com.001
    </individualCredentialHandle>
    <Name>Leslie L. Daigle</Name>
    <individualID>leslie@thinkingcat.com</individualID>
    <domainName>thinkingcat.com</domainName>
    <organizationName>Thinking Cat Enterprises</organizationName>
    <credentialApplication>any</credentialApplication>
    <publicKey>
ssh-dss AAAAB3NzaC1kc3MAAABBAJMPauNLsiZOpsp2bWEzXrP/TDeOPuVTuK/xzXYPGNpw2gU/
6DDwP9Ulb64KfG3aV3L8mxbquRiCIMQ04EbR0/
EAAAAVAJP8mw55riJWZfr13aoFjE9mMuRzAAAAQQCNQ7r994sofueXgVequqBCgHPWBtJm5wX1VUvy4mBLwwaEut7algsl8
TG1mql3nH8Y2HJKkQaqO/4= leslie@thinkingcat.com
    </publicKey>
  </individualCredential>
```

4.2.2 <findIndividualByName> query

<findIndividualByName> finds an individual credential entity by the name of the individual. Optional search constraints include: <domainName> to specify the results should be in domains within the domain specified by its content; <Organization> to specify the target organization; and <credentialApplication> to specify the target application of the credential. The query may also specify that the results should be presented as entity references only (<entityRefsOnly>).

The <findIndividualByName&gt query returns a result set of <individualCredential> or of <entityRefResult> if the client requested entity references only. These are defined in Section 4.1.

Query fragment

```
<findIndividualByName>
  <Name>
    <endsWith>Daigle</endsWith>
  </Name>
  <domainName>thinkingcat.com</domainName>
</findIndividualByName>
```

Response fragment

```
<individualCredential>
  <individualCredentialHandle>
    leslie.thinkingcat.com.001
  </individualCredentialHandle>
  <Name>Leslie L. Daigle</Name>
  <individualID>leslie@thinkingcat.com</individualID>
  <domainName>thinkingcat.com</domainName>
  <organizationName>Thinking Cat Enterprises</organizationName>
  <credentialApplication>any</credentialApplication>
  <publicKey>
ssh-dss AAAAB3NzaC1kc3MAAABBAJMPauNLsiZOpsp2bWEzXrP/TDeOPuVTuK/xzXYPGNpw2gU/
6DDwP9Ulb64KfG3aV3L8mxbquRiCIMQ04EbR0/
EAAAAVAJP8mw55riJWZfr13aoFjE9mMuRzAAAAQQCNQ7r994sofueXgVequqBCgHPWBtJm5wX1VUvy4mBLwwaEut7algsl8
TG1mql3nH8Y2HJKkQaqO/4= leslie@thinkingcat.com
  </publicKey>
</individualCredential>

<individualCredential>
  <individualCredentialHandle>
      leslie.thinkingcat.com.003
  </individualCredentialHandle>
  <Name>Leslie L. Daigle</Name>
  <individualID>leslie@thinkingcat.com</individualID>
  <domainName>jabber.thinkingcat.com</domainName>
  <organizationName>Thinking Cat Enterprises</organizationName>
  <credentialApplication>IM</credentialApplication>
  <credentialURI>
      http://pgp.mit.edu:11371/pks/lookup?op=get&search=0x5C568519
  </credentialURI>
</individualCredential>
```

4.2.3 <findServerByName> query

The <findServerByName> is used for finding credentials of servers (e.g., web servers). An optional search constraint of <credentialApplication> can be used to indicate the particular application for which the credential is sought. A single machine (name) may host several different application servers, with different credentials. The client may request that the results contain entity references only (<entityRefsOnly>).

The <findServerByName&gt query returns a result set of <serverCredential> or of <entityRefResult> if the client requested entity references only. These are defined in Section 4.1.

Query fragment

```
<findServerByName>
  <Name>thinkingcat.com</Name>
</findServerByName>
```

Response fragment

```
<serverCredential>
  <serverCredentialHandle>
     thinkingcat.com.010
  </serverCredentialHandle>
  <serverName>www.thinkingcat.com</serverName>
  <domainName>thinkingcat.com</domainName>
  <organizationName>
      Thinking Cat Enterprises
  </organizationName>
  <credentialApplication>www</credentialApplication>
  <publicKey attachment="1"/>
</serverCredential>
```

4.2.4 Support for <iris:lookupEntity>

Two types of named entities are supported for the <lookupEntity> query:

- <individual> for looking up credentials for individuals. A result of <individualCredential> is returned.

- <server> for looking up credentials for servers. A result of

<serverCredential> is returned.

5. Formal XML Syntax

This registry schema is specified in the XML Schema notation. The formal syntax presented here is a complete schema representation suitable for automated validation of an XML instance when combined with the formal schema syntax of IRIS.

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:credreg="urn:ietf:params:xml:ns:credreg1"
        xmlns:iris="urn:ietf:params:xml:ns:iris1"
        targetNamespace="urn:ietf:params:xml:ns:credreg1"
        elementFormDefault="qualified" >

  <import namespace="urn:ietf:params:xml:ns:iris1" />

  <annotation>
    <documentation>
      (Security) Credentials Registry schema
      derived from IRIS schema

      Entity classes

      These are split up as:  individual/server.  Another
      cut would be to make it one per application type
      (per individual/server).  The disadvantage to the
      latter is having to set the list of applications
      in the actual schema spec.  The disadvantage to
      this way is that the individual identifier is
      not a (database) key, so one must be generated,
      and the fact that mayhem may ensue (since there
      is no standard set of credential application types).


      Supported result types:

        individualCredential
          individualCredentialHandle (required)
          Name (optional)
          Role (optional)
          individualID (optional)
          domainName (optional)
          credentialApplication (required)
          One of publicKey (optional), publicCert (optional),
             credentialURI (optional)
```

```
          serverCredential
            serverCredentialHandle (required)
            serverName (required)
            domainName  (optional)
            organizationName (optional)
            credentialApplication (required)
            One of:  publicKey (optional), publicCert (optional)
            credentialURI (optional)


          entityRefResult
            thisEntityURI (required)


        Supported query types:
          findIndividualByID
            Input (now):
              Required: ID (exact or beginning)
              Optional: credential application
           Optional: request entity references only
            Considered refinements:
              Organization name
              (Individual) name

          findIndividualByName
            Input (now):
              Required: Name (exact, beginning, or ending)
              Optional: DomainName (ending)
              Optional: Organization (beginning)
              Optional: credential application
           Optional: request entity references only
            Considered refinements:
              <none>

          findServerByName
            Input (now):
              Required:  server name (ID)
              Optional:  credential application
           Optional:  request entity references only
            Considered refinements:
           Domain Name (name of the domain this server serves)
              Organization name


      </documentation>
    </annotation>
```

```
    <!--             -->
    <!-- Query types -->
    <!--             -->

    <complexType name="findIndividualByID-Type">
      <complexContent>
        <extension base="iris:queryType">
          <choice>
            <element name="beginsWith">
              <simpleType>
                <restriction base="token">
                  <minLength value="3"/>
                </restriction>
              </simpleType>
            </element>
            <element name="exactMatch"
              type=normalizedString" />
          </choice>
          <element name="credentialApplication"
            type="token" minOccurs="0" maxOccurs="1" />
          <element name="entityRefsOnly"
            type="boolean" default="false" minOccurs="0" maxOccurs="1" />
        </extension>
      </complexContent>
    </complexType>

    <element name="findIndividualByID"
      type="credreg:findIndividualByID-Type"
      substitutionGroup="iris:query" />


    <complexType name="findIndividualByName-Type">
      <complexContent>
        <extension base="iris:queryType">
          <element name="Name">
            <complexType>
              <choice>
                <element name="beginsWith">
                  <simpleType>
                    <restriction base="normalizedString">
                      <minLength value="3"/>
                    </restriction>
                  </simpleType>
                </element>
                <element name="endsWith">
                  <simpleType>
                    <restriction base="normalizedString">
                      <minLength value="3"/>
```

```
                    </restriction>
                  </simpleType>
                </element>
                <element name="exactMatch"
                  type=normalizedString" />
              </choice>
            <complexType>
          </element>
          <element name="credentialApplication"
            type="token" minOccurs="0" maxOccurs="1" />
          <element name="DomainName" minOccurs="0" maxOccurs="1">
            type="token" />
          <element name="Organization" minOccurs="0" maxOccurs="1">
            type="normalizedString" />
          <element name="entityRefsOnly"
            type="boolean" default="false" minOccurs="0" maxOccurs="1" />
        </extension>
      </complexContent>
    </complexType>

    <element name="findIndividualByName"
      type="credreg:findIndividualByName-Type"
      substitutionGroup="iris:query" />


    <complexType name="findServerByName-Type">
      <complexContent>
        <extension base="iris:queryType">
          <element name="Name" type="normalizedString">
          <element name="credentialApplication"
            type="token" minOccurs="0" maxOccurs="1" />
          <element name="entityRefsOnly"
            type="boolean" default="false" minOccurs="0" maxOccurs="1" />
        </extension>
      </complexContent>
    </complexType>

    <element name="findServerByName"
      type="credreg:findServerByName-Type"
      substitutionGroup="iris:query" />



    <!--              -->
    <!-- Result types -->
    <!--              -->

    <complexType name="individualCredential-Type">
```

```
      <complexContent>
        <extension base="iris:resultType">
          <sequence>
            <element name="individualCredentialHandle"
              type="normalizedString" />
            <element name="Name"
              type="string"
              minOccurs="0" maxOccurs="1" />
            <element name="Role"
              type="string"
              minOccurs="0" maxOccurs="1" />
            <element name="individualID"
              type="normalizedString"
              minOccurs="0" maxOccurs="1" />
            <element name="domainName"
              type="normalizedString"
              minOccurs="0" maxOccurs="1" />
            <element name="organizationName"
              type="string"
              minOccurs="0" maxOccurs="1" />
            <element name="credentialApplication"
              type="token" default="any"
              minOccurs="1" maxOccurs="1" />
            <choice>
              <element name="publicKey"
                type="string" />
              <element name="publicCert"
                type="string" />
            </choice>
            <element name="credentialURI"
              type="anyURI"
              minOccurs="0" maxOccurs="1" />
          </sequence>
        </extension>
      </complexContent>
    </complexType>

    <element name="individualCredential"
      type="credreg:individualCredential-Type"
      substitutionGroup="iris:result" />

    <complexType name="serverCredential-Type">
      <complexContent>
        <extension base="iris:resultType">
          <sequence>
            <element name="serverCredentialHandle"
              type="normalizedString" />
            <element name="serverName"
```

```
              type="normalizedString" />
            <element name="domainName"
              type="normalizedString"
              minOccurs="0" maxOccurs="1" />
            <element name="organizationName"
              type="string"
              minOccurs="0" maxOccurs="1" />
            <element name="credentialApplication"
              type="token" default="any" />
            <choice>
              <element name="publicKey"
                type="string"
                minOccurs="0" maxOccurs="1" />
              <element name="publicCert"
                type="string"
                minOccurs="0" maxOccurs="1" />
            </choice>
            <element name="credentialURI"
              type="anyURI"
              minOccurs="0" maxOccurs="1" />
          </sequence>
        </extension>
      </complexContent>
    </complexType>

    <element name="serverCredential"
      type="credreg:serverCredential-Type"
      substitutionGroup="iris:result" />


    <element name="entityRefResult"
      type="iris:resultType"
      substitutionGroup="iris:result" />


    <!--             -->
    <!-- Error types -->
    <!--             -->

    <element name="searchTooWide"
      type="iris:codeType"
      substitutionGroup="iris:genericCode" />


  </schema>
```

6. credreg and IRIS-lw

The credreg service may be supported using the lightweight UDP IRIS service, IRIS-lw ([7]). In this case, only the <iris:lookupEntity> and searches with <entityRefsOnly> set to "true" are supported via IRIS-lw.

7. credreg Server Location Convention

Although individual credreg servers are operated independently, there is a convention for locating authoritative servers for credential material. This convention uses the specific application of NAPTR DNS resource records defined in [6].

This document defines the "credreg" application service. Clients seeking authoritative credreg servers for credential materials should look up the credreg service in the domain of application of the credential material (the domain of the named server, the domain in the e-mail address, etc).

For example, to find the credentials associated with the e-mail address "someone@example.com", the NAPTR records for the credreg application service in the "example.com" domain are looked up.

```
example.com.
;;      order   pref    flags   service regexp  replacement
IN NAPTR 100    10      "s"     "credreg+iris" ""    _iris._tcp.someisp.com.
```

and then the administrators at someisp.com can manage the preference rankings of the servers they use to support the prim service:

```
_credreg._tcp.someisp.com.
 ;;    Pref Weight Port  Target
IN SRV 10    0     10001 bigiron.example.com
IN SRV 20    0     10001 backup.im.example.com
IN SRV 30    0     10001 nuclearfallout.example.com.au
```

8. Internationalization Considerations

Implementers should be aware of considerations for internationalization in IRIS [5].

9. IANA Considerations

The following URN will need to be registered with IANA according to the IANA considerations defined in IRIS [5]:

urn:ietf:params:xml:ns:credreg1

Will also need to register the "credreg" service type for the NAPSTR server location scheme.

10. Security Considerations

This document lays out no new considerations for security precautions beyond that specified in IRIS [5].

11. Acknowledgements

The author would like to thank Andy Newton for his input and many spirited discussions on the topic. As I set down my virtual pen and declare this rev complete, it is with anticipation of many more spirited discussions, to the benefit of future revisions :-)

References

[1] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, February 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>.

[2] World Wide Web Consortium, "Namespaces in XML", W3C XML Namespaces, January 1999, <http://www.w3.org/TR/1999/REC-xml-names-19990114>.

[3] World Wide Web Consortium, "XML Schema Part 2: Datatypes", W3C XML Schema, October 2000, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>.

[4] World Wide Web Consortium, "XML Schema Part 1: Structures", W3C XML Schema, October 2000, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>.

[5] Newton, A., "Internet Registry Information Service", draft-ietf-crisp-iris-core-01 (work in progress), November 2002.

[6] Daigle, L. and A. Newton, "Domain-based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", draft-daigle-napstr-01 (work in progress), November 2002.

[7] Newton, A. and L. Daigle, "Lightweight Internet Registry Information Service", draft-newtion-iris-lightweight-00 (work in progress), February 2003.

[8] Reynolds, J. and J. Postel, "ASSIGNED NUMBERS", RFC 1700, STD 2, October 1994.

[9] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, BCP 26, October 1998.

[10] Newton, A., "Cross Registry Internet Service Protocol (CRISP) Requirements", draft-ietf-crisp-requirements-00 (work in progress), August 2002.

[11] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[12] World Wide Web Consortium, "XML Key Management Specification (XKMS)", W3C XKMS, March 2001, <http://www.w3.org/TR/2001/NOTE-xkms-20010330/>.

Author's Address

Leslie L. Daigle
VeriSign, Inc.
21345 Ridgetop Circle
Sterling, VA 20166
USA

Phone: +1 703 948 3385
EMail: leslie@verisignlabs.com
URI: http://www.verisignlabs.com/

Appendix A. Complete Example Request and Response

The following is a complete example of an IRIS request and response using this registry schema.

This XML instance is a request to search for an individual by a portion of the individual's ID.

```
<?xml version="1.0"?>
<iris xmlns="urn:ietf:params:xml:ns:iris1"
      xmlns:iris="urn:ietf:params:xml:ns:iris1"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:ietf:params:xml:ns:iris1 iris.xsd" >

  <request>
    <searchSet>
      <credreg:findIndividualByID
        xmlns:credreg="urn:ietf:params:xml:ns:credreg1"
        xsi:schemaLocation="urn:ietf:params:xml:ns:credreg1 credreg.xsd" >
        <credreg:beginsWith>leslie</beginsWith>
      </credreg:findIndividualByID>
    </searchSet>
  </request>

</iris>
```

This XML instance is a response from Figure 7.

```
<?xml version="1.0"?>
<iris xmlns="urn:ietf:params:xml:ns:iris1"
      xmlns:iris="urn:ietf:params:xml:ns:iris1"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:ietf:params:xml:ns:iris1 iris.xsd" >

  <response>
    <resultSet>
      <answer>
        <credreg:individualCredential
          xmlns="urn:ietf:params:xml:ns:credreg1"
          xmlns:credreg="urn:ietf:params:xml:ns:credreg1"
          xsi:schemaLocation="urn:ietf:params:xml:ns:credreg1 credreg.xsd"
          thisEntityURI="iris://thinkingcat.com/credreg1/individual/
leslie.thinkingcat.com.001">
            <credreg:individualCredentialHandle>
               leslie.thinkingcat.com.001
            </credreg:individualCredentialHandle>
            <credreg:Name>Leslie L. Daigle</credreg:Name>
            <credreg:individualID>leslie@thinkingcat.com</
credreg:individualID>
            <credreg:domainName>thinkingcat.com</credreg:domainName>
            <credreg:organizationName>Thinking Cat Enterprises</
credreg:organizationName>
            <credreg:credentialApplication>any</
credreg:credentialApplication>
            <credreg:publicKey>
   ssh-dss AAAAB3NzaC1kc3MAAABBAJMPauNLsiZOpsp2bWEzXrP/TDeOPuVTuK/xzXYPGNpw2gU/
6DDwP9Ulb64KfG3aV3L8mxbquRiCIMQ04EbR0/
EAAAAVAJP8mw55riJWZfr13aoFjE9mMuRzAAAAQQCNQ7r994sofueXgVequqBCgHPWBtJm5wX1VUvy4mBLwwaEut7algsl8
TG1mql3nH8Y2HJKkQaqO/4= leslie@thinkingcat.com
            </credreg:publicKey>
          </credreg:individualCredential>
        </answer>
      </resultSet>
    </response>

  </iris>
```

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.