                  ENUM Validation Token Format Definition

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Abstract

   An ENUM domain name is tightly coupled with the underlying E.164
   number. The process of verifying whether the Registrant of an ENUM
   domain name is identical to the Assignee of the corresponding E.164
   number is commonly called "validation". This document describes a
   signed XML data format -- the Validation Token -- with which
   Validation Entities can convey successful completion of a validation
   procedure in a secure fashion.

Table of Contents

1. Introduction

   In the case where an ENUM (E.164 Number Mapping [1]) domain name
   corresponds to an existing E.164 number [2], the delegation of this
   domain needs to be authorized by the Assignee of the corresponding
   E.164 number. In the role model described in [15], the entity that
   performs this check is called the Validation Entity (VE).

   By conveying an ENUM Validation Token -- a signed XML document -- to
   the Registry, a VE certifies that delegation requirements have been
   met and are current.

   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 RFC 2119 [3].

2. Data Requirements

   In this model, the Token is the only piece of data passed from the VE
   to the Registry. Therefore, the Token needs to contain at least as
   much information as the Registry requires to grant the delegation of
   the requested ENUM domain according to its registration policy. As
   such, the Registry will need confirmation that:

   o the Token was created by an accredited VE,

   o the Token's duration of validity conforms to the policy,

   o the validation procedure employed has met minimum requirements as
      set forth by policy,

   o and that the Token is protected against tampering and replay
      attacks.

   Beyond such mandatory information, the Token may optionally include
   number holder information, in particular, to simplify future
   revalidations.

   For example, if initial validation requires the steps "Check the
   identity of the Registrant" and "Check the ownership of an E.164
   number", then a later revalidation only needs to re-check the
   ownership as the identity of the Registrant does not change.

   As the Token will be included (see e.g., [16]) in XML-based Registry/
   Registrar protocols like the Extensible Provisioning Protocol (EPP)
   [13], it is a natural choice to use XML to encode Validation Tokens.

3. Digital Signature

   According to the architecture model the propriety of an ENUM
   delegation depends on the trust relationship between the Registry and
   the VE. In general, an untrusted link between the Registry and VE
   should be assumed (for instance, the Token is passed along with the
   registration request by a Registrar, who might have no role in
   asserting the right-to-use). Therefore, the Token must be protected
   against forgery, tampering, and replay-attacks.

   A digital signature on the token:

   o asserts that the token was indeed generated by the indicated VE
      (authenticity).

   o guarantees that the token was not tampered with in transit
      (integrity).

   o enables auditing the validation process (non-repudiation).

   The cryptographic signature on the token follows RFC 3275 (XML-DSIG
   [4]). As tokens might be transmitted as part of an already XML based
   protocol, the exclusive XML canonicalization [9] MUST be used. This
   transform guarantees that namespace declarations inherited from the
   surrounding XML do not invalidate the signature. In order to make
   the signature an integral part of the token, the
   "enveloped"-signature mode is employed. The signature covers all
   information contained in the Token.

   XML-DSIG offers a number of cryptographic algorithms for digesting
   and signing documents and recommends SHA1/RSA-SHA1. Recent advances
   in cryptanalysis have cast doubt on the security of SHA1, thus
   rendering this recommendation obsolete (see e.g., the Security
   Considerations of [14]). RFC 4051 [5] defines how additional
   algorithms can be used with XML-DSIG.

   Validation Entities MUST be able to sign tokens according to
   XML-DSIG, MUST support RSA-SHA1 and RSA-SHA256 [5], MUST support RSA
   key sizes of 1024 and 2048 bits, and MUST be able to embed X.509 [10]
   certificates. The Registry MUST define which signature algorithms
   and key sizes it will accept in Validation Tokens as part of its
   local policy.

   The choice of a RSA-based signature does not require a public key
   infrastructure. Whether the Registry acts as a certification
   authority, accepts certs from a public certification authority, or
   only accepts pre-registered keys is a local policy choice.

4. Field Descriptions

   The Validation Token is structured into three parts: the basic
   validation information, additional information about the Registrant,
   and the digital signature. The XML schema can be found in Section 6.

4.1. The <validation> Element

   A token MUST contain a <validation> element that contains the
   following:

   o A single validation "serial" attribute identifying a validation
      token for a certain VE. It must be unique per VE.

   o A single <E164Number> element containing the underlying E.164
      number in fully qualified (international) format.

   o An optional <lastE164Number> element. If present, it indicates
      that the whole number block starting with <E164Number> up to and
      including <lastE164Number> has been validated. To avoid
      ambiguity, both numbers MUST be of the same length.

   o A single <validationEntityID> element identifying the VE.

   o A single <registrarID> element identifying the Registrar on whose
      behalf the validation was performed.

   o A single <methodID> element identifying the method used by the VE
      for validation.

   o A single <executionDate> attribute containing the date of
      validation formatted as "full-date" according to RFC 3339 [6].

   o An optional <expirationDate> attribute marking the expiration date
      of the validation token formatted as "full-date" according to RFC
      3339. The Registry will automatically revoke the delegation at
      this date unless a new Token has been submitted that extends the
      lifetime of the validation. A missing <expirationDate> indicates
      infinite validity of the Token.

   The format and the uniqueness-constraints of these IDs is left to the
   local policy of the Registry.

4.2. The <tokendata> Element

   A token may contain a <tokendata> section containing information
   about the number holder, consisting of the following elements:

   o A single <organization> element containing the full name of the
      organization to which the Registrant is affiliated.

   o A single <commercialregisternumber> element. If the Registrant is
      a company, then this field can be used to uniquely identify this
      company by its official registration number within the local
      country. The interpretation of this field is thus
      country-specific.

   o A single <title> element.

   o A single <firstname> element.

   o A single <lastname> element.

   o A single <address> section containing the following elements:
      * A single optional <streetName>
      * A single optional <houseNumber>
      * A single optional <postalCode>
      * A single optional <locality>
      * A single optional <countyStateOrProvince>
      * A single optional <ISOcountryCode>

   o Up to 10 <phone> elements containing full E.164 numbers.

   o Up to 10 <fax> elements containing full E.164 numbers.

   o Up to 10 <email> elements.

   All elements directly under <tokendata> are optional. The
   <ISOcountryCode> element specifies the country using the alpha-2
   country code from ISO 3166-1:2006 [11] (including updates published
   by the 3166 Maintenance Agency). The definition of the first five
   elements within the <address> element conforms to the second version
   of the E.115 Computerized Directory Assistance [17].

5. Examples

5.1. Unsigned Token without Registrant Information

   This basic Token without any information about the Registrant and
   without the cryptographic signature shows the basic layout of the
   Token.

   ```
   <?xml version="1.0" encoding="utf-8" standalone="no" ?>
   <token xmlns="urn:ietf:params:xml:ns:enum-token-1.0" Id="TOKEN"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation=
    "urn:ietf:params:xml:ns:enum-token-1.0 enum-token-1.0.xsd">
     <validation serial="acmeve-000002">
       <E164Number>+442079460200</E164Number>
       <lastE164Number>+442079460499</lastE164Number>
       <validationEntityID>ACME-VE</validationEntityID>
       <registrarID>reg-4711</registrarID>
       <methodID>42</methodID>
       <executionDate>2007-05-08</executionDate>
       <expirationDate>2007-11-01</expirationDate>
     </validation>
   </token>
   ```

5.2. Signed Token

   This example uses an X.509 based signature that includes the
   certificate of the signing validation entity. Thus, the validity of
   the signature can be verified without the need for a key-server. A
   valid signature is a necessary, but not sufficient, condition for a
   valid Token. Any entity evaluating a Token needs to check other
   factors as well, e.g., the certificate and the XML schema.

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<token xmlns="urn:ietf:params:xml:ns:enum-token-1.0" Id="TOKEN"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation=
 "urn:ietf:params:xml:ns:enum-token-1.0 enum-token-1.0.xsd">
  <validation serial="acmeve-000001">
    <E164Number>+442079460123</E164Number>
    <validationEntityID>ACME-VE</validationEntityID>
    <registrarID>reg-4711</registrarID>
    <methodID>42</methodID>
    <executionDate>2007-05-08</executionDate>
  </validation>
  <tokendata xmlns="urn:ietf:params:xml:ns:enum-tokendata-1.0"
   xsi:schemaLocation=
   "urn:ietf:params:xml:ns:enum-tokendata-1.0 enum-tokendata-1.0.xsd">
```

```
        <contact>
          <organisation>Example Inc.</organisation>
          <commercialregisternumber>4711</commercialregisternumber>
          <title>Dr.</title>
          <firstname>Max</firstname>
          <lastname>Mustermann</lastname>
          <address>
            <streetName>Main</streetName>
            <houseNumber>10</houseNumber>
            <postalCode>1010</postalCode>
            <locality>London</locality>
            <countyStateOrProvince>London</countyStateOrProvince>
            <ISOcountryCode>GB</ISOcountryCode>
          </address>
          <phone>+442079460123</phone>
          <email>mm@example.com</email>
        </contact>
      </tokendata>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod
           Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          <SignatureMethod
           Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
          <Reference URI="#TOKEN">
            <Transforms>
              <Transform Algorithm=
               "http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
              <Transform
               Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <InclusiveNamespaces
                 xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"
                 PrefixList="enum-token enum-tokendata"/>
              </Transform>
            </Transforms>
            <DigestMethod
             Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <DigestValue
             >VxqsBxSNPFwPAUlCHts3g3DehcexnB1dqUz+GypLZ0k=</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>
QKqphKRNPokVZFbenje+HZZV+RLrNweGnlWBw7ngAtH+rtuslR8LhMLmC4DlBb9V
HvKItl+7zLGm3VgYsqfHH8q3jCl1mFxUIuLlIPqtpJs+xAHAJDzZ+vmsF/q2IgrS
K0uMmKuU5V1gydDBOvIipcJx+PrPYyXYZSjQXkWknK8=</SignatureValue>
      <KeyInfo>
<X509Data>
<X509Certificate>
```

MIIDZjCCAs+gAwIBAgIBBDANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJBVDEP
MA0GA1UEBxMGVmllbm5hMRQwEgYDVQQKEwtCT0ZIIENlcnRzLjEbMBkGA1UEAxMS
Q0VSVFMuYm9maC5wcml2LmF0MSEwHwYJKoZIhvcNAQkBFhJjZXJ0c0Bib2ZoLnBy
aXYuYXQwHhcNMDQwNzIwMTMxNTA5WhcNMDUwNzIwMTMxNTA5WjB/MQswCQYDVQQG
EwJBVDEKMAgGA1UECBMBLTEPMA0GA1UEBxMGVmllbm5hMR0wGwYDVQQKExRBY21l
IEVOVU0gVmFsaWRhdGlvbjEQMA4GA1UEAxMHYWNtZS1WRTEiMCAGCSqGSIb3DQEJ
ARYTbm9ib2R5QGVudW0tYWNtZS5hdCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkC
gYEArJPcjMFc54/zwztSdQXGxUtodJT9r1qGI2lQPNjLvtPJg93+7o5SIOsZGSpg
zWbztDAV5qc7PHZWUVIyf6MbM5qSgQDVrjNRhTosNtyqmwi23BH52SKkX3P7eGit
LmqEkiUZRxZhZ6upRbtcqvKSwmXitvW4zXZhkVHYJZ2HuMcCAwEAAaOB/DCB+TAJ
BgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0
aWZpY2F0ZTAdBgNVHQ4EFgQUyK4otTQtvv6KdSlMBOPT5Ve18JgwgZ4GA1UdIwSB
ljCBk4AUvfPadpm0HhmZx2iAVumQTwgnG2eheKR2MHQxCzAJBgNVBAYTAkFUMQ8w
DQYDVQQHEwZWaWVubmExFDASBgNVBAoTC0JPRkggQ2VydHMuMRswGQYDVQQDExJD
RVJUUy5ib2ZoLnByaXYuYXQxITAfBgkqhkiG9w0BCQEWEmNlcnRzQGJvZmgucHJp
di5hdIIBADANBgkqhkiG9w0BAQQFAAOBgQCB9CHBnIUhrdic4h5Ar4hdxjHSQkDH
sJWd+MYrNcuSrv3TIOsUkUgNpNNhmkZPtiXqfy3388IRdJtJiLWXSOb/XlZHOM9I
MvwKYwhcpQ9UdM/w7VpXQqf+CEj0XSyqxGw65UsHIOijgiG/WyhSj+Lzriw7CTge
P2iAJkJVC4t2XA==
</X509Certificate>
</X509Data>
</KeyInfo>
</Signature>
</token>

6. Formal Syntax

    The formal syntax of the validation token is specified using XML
    schema notation [7] [8]. Two schemas are defined: The "token core
    schema" contains mandatory attribute definitions, and the "token data
    schema" defines the format of the optional "tokendata" section. The
    BEGIN and END tags are not part of the schema; they are used to note
    the beginning and ending of the schema for URI registration purposes.

6.1. Token Core Schema

```
   BEGIN
   <?xml version="1.0" encoding="UTF-8"?>

   <schema targetNamespace="urn:ietf:params:xml:ns:enum-token-1.0"
     xmlns:enum-token="urn:ietf:params:xml:ns:enum-token-1.0"
     xmlns:enum-tokendata="urn:ietf:params:xml:ns:enum-tokendata-1.0"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
     xmlns="http://www.w3.org/2001/XMLSchema"
     elementFormDefault="qualified">

   <!--  Import common element types.  -->

     <import namespace="http://www.w3.org/2000/09/xmldsig#"
             schemaLocation="xmldsig-core-schema.xsd"/>
     <import namespace="urn:ietf:params:xml:ns:enum-tokendata-1.0"
             schemaLocation="enum-tokendata-1.0.xsd"/>

     <annotation>
       <documentation>
         Validation Token core schema
       </documentation>
     </annotation>

     <element name="token" type="enum-token:tokenBaseType"/>

     <simpleType name="shortTokenType">
       <restriction base="token">
         <minLength value="1"/>
         <maxLength value="20"/>
       </restriction>
     </simpleType>

     <simpleType name="e164numberType">
       <restriction base="token">
         <maxLength value="20"/>
         <pattern value="\+\d\d*"/>
       </restriction>
     </simpleType>

     <complexType name="validationDataType">
       <sequence>
         <element name="E164Number"
                       type="enum-token:e164numberType"/>
         <element name="lastE164Number" minOccurs="0"
                       type="enum-token:e164numberType"/>
         <element name="validationEntityID"
```

```
                            type="enum-token:shortTokenType"/>
        <element name="registrarID"
                            type="enum-token:shortTokenType"/>
        <element name="methodID"
                            type="enum-token:shortTokenType"/>
        <element name="executionDate" type="date"/>
        <element name="expirationDate"
                            type="date" minOccurs="0"/>
      </sequence>
      <attribute name="serial" type="enum-token:shortTokenType"
       use="required"/>
    </complexType>

    <complexType name="tokenBaseType">
      <sequence>
        <element name="validation"
         type="enum-token:validationDataType"/>
        <any namespace="urn:ietf:params:xml:ns:enum-tokendata-1.0"
         minOccurs="0"/>
        <any namespace="http://www.w3.org/2000/09/xmldsig#"/>
      </sequence>
      <attribute name="Id" type="ID" use="required"/>
    </complexType>
  </schema>
  END
```

6.2. Token Data Schema

```
  BEGIN
  <?xml version="1.0" encoding="UTF-8"?>

  <schema targetNamespace="urn:ietf:params:xml:ns:enum-tokendata-1.0"
    xmlns:enum-tokendata="urn:ietf:params:xml:ns:enum-tokendata-1.0"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <element name="tokendata" type="enum-tokendata:tokenDataType"/>

    <simpleType name="E115String">
      <restriction base="string">
   <pattern value="[&#x20;-&#x7A; -&#xD7FF;&#xE000;-&#xFFFD;]*"/>
      </restriction>
    </simpleType>

    <simpleType name="E115StringUb256">
      <restriction base="enum-tokendata:E115String">
        <minLength value="1"/>
        <maxLength value="256"/>
```

```
        </restriction>
      </simpleType>

      <simpleType name="countryCodeType">
        <restriction base="token">
          <minLength value="2"/>
          <maxLength value="2"/>
        </restriction>
      </simpleType>

      <simpleType name="TokenType">
        <restriction base="token">
          <minLength value="1"/>
          <maxLength value="64"/>
        </restriction>
      </simpleType>

      <complexType name="addressType">
        <all>
          <element name="streetName"      minOccurs="0"
           type="enum-tokendata:E115StringUb256" />
          <element name="houseNumber"     minOccurs="0"
           type="enum-tokendata:E115StringUb256"/>
          <element name="postalCode"      minOccurs="0"
           type="enum-tokendata:E115StringUb256"/>
          <element name="locality"        minOccurs="0"
           type="enum-tokendata:E115StringUb256"/>
          <element name="countyStateOrProvince" minOccurs="0"
           type="enum-tokendata:E115StringUb256"/>
          <element name="ISOcountryCode" minOccurs="0"
           type="enum-tokendata:countryCodeType"/>
        </all>
      </complexType>

      <group name="tokenContactBaseGroup">
        <sequence>
          <element name="organisation"  minOccurs="0"
           type="enum-tokendata:E115StringUb256"/>
          <element name="commercialregisternumber" minOccurs="0"
           type="enum-tokendata:TokenType"/>
          <element name="title"           minOccurs="0"
           type="enum-tokendata:TokenType"/>
          <element name="firstname"      minOccurs="0"
           type="enum-tokendata:E115StringUb256"/>
          <element name="lastname"        minOccurs="0"
           type="enum-tokendata:E115StringUb256"/>
          <element name="address"         minOccurs="0"
           type="enum-tokendata:addressType"/>
```

```
         <element name="phone" type="enum-tokendata:TokenType"
          minOccurs="0" maxOccurs="10" />
         <element name="fax"   type="enum-tokendata:TokenType"
          minOccurs="0" maxOccurs="10" />
         <element name="email" type="enum-tokendata:TokenType"
          minOccurs="0" maxOccurs="10" />
       </sequence>
     </group>

     <complexType name="contactType">
       <sequence>
         <group ref="enum-tokendata:tokenContactBaseGroup"/>
       </sequence>
     </complexType>

     <complexType name="tokenDataType">
       <sequence>
         <element name="contact" type="enum-tokendata:contactType"/>
       </sequence>
     </complexType>

   </schema>
   END
```

7. Other Applications of the Token Concept

The concept of the validation token may be useful in other
registry-type applications where the proof of an underlying right is
a condition for a valid registration.

An example is a Top Level Domain (TLD) where registration is subject
to proof of some precondition, like a trade mark or the right in a
name. Such situations often arise during the introduction of a new
TLD, e.g., during a "sunrise" phase.

A Number Portability (NP) database faces very similar verification
issues. An NP system based on the Token concept could potentially be
superior to current methods, and aid in the convergence of NP and
ENUM.

8. IANA Considerations

This document uses Uniform Resource Names (URNs) to describe XML
namespaces and XML schemas conforming to a registry mechanism
described in RFC 3688 [12]. IANA has made the following four URI
assignments.

   1. Registration for the Token namespace:
       *   URI: urn:ietf:params:xml:ns:enum-token-1.0
       *   Registrant Contact: See the "Author's Address" section of this
           document.
       *   XML: None. Namespace URIs do not represent an XML
           specification.

   2. Registration for the Token XML schema:
       *   URI: urn:ietf:params:xml:schema:enum-token-1.0
       *   Registrant Contact: See the "Author's Address" section of this
           document.
       *   XML: See Section 6.1 of this document.

   3. Registration for the Token Data namespace:
       *   URI: urn:ietf:params:xml:ns:enum-tokendata-1.0
       *   Registrant Contact: See the "Author's Address" section of this
           document.
       *   XML: None. Namespace URIs do not represent an XML
           specification.

   4. Registration for the Token Data XML schema:
       *   URI: urn:ietf:params:xml:schema:enum-tokendata-1.0
       *   Registrant Contact: See the "Author's Address" section of this
           document.
       *   XML: See Section 6.2 of this document.

   The IDs used in the validationEntityID, RegistrarID, and methodID
   elements are subject to local policy and thus do not require IANA
   registration.

9. Security Considerations

   The security of the Validation Token depends on the security of the
   underlying XML DSIG algorithms. As such, all the security
   considerations from [4] apply here as well. Two points from [4]
   merit repetition:

   Transforms are used to select the relevant data for signing and
   discarding irrelevant information (e.g., pretty-printing and
   name-space local names).

   The <Reference URI="#TOKEN"> element and attribute combined with the
   Id="TOKEN" attribute in <token> specifies that the signature should
   cover the complete token. Moving the Id="TOKEN" attribute to e.g.,
   the <tokendata> element would make the signature worthless.

It is thus critical that the Registry not only checks whether the
Token passes a generic XML-DSIG signature check, but also that:

1. the signature uses approved transforms and cryptographic
    algorithms.
2. the signature references the <token> element.
3. the key used in the signature belongs to an accredited VE.

The Token content is not encrypted. If local policy dictates that
the information contained within the token should be confidential,
then this has to be handled through a different mechanism.

When processing a delegation request, the Registry MUST verify that
the information contained in the Token matches the delegation
request. The <registrarID> element in the Token prevents a malicious
second Registrar from using an eavesdropped Token to register a
domain in his name. The Registry MUST verify that the
<expirationDate> given (including the case of no given expiration
date) conforms to the Registry's policy. To avert replay attacks,
local policy MUST specify how long after <executionDate> the Token
can be used to authorize a delegation.

10. Acknowledgements

The author would like to thank the following persons for their
valuable suggestions and contributions: Michael Haberler, Alexander
Mayrhofer, Bernie Hoeneisen, Michael Braunoeder, Staffan Hagnell,
Lawrence Conroy, and Tony Rutkowski.

11. References

11.1. Normative References

[1]   Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource
        Identifiers (URI) Dynamic Delegation Discovery System (DDDS)
        Application (ENUM)", RFC 3761, April 2004.

[2]   ITU-T, "The international public telecommunication numbering
        plan", Recommendation E.164, May 1997.

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

[4]   Eastlake 3rd, D., Reagle, J., and D. Solo, "(Extensible Markup
        Language) XML-Signature Syntax and Processing", RFC 3275, March
        2002.

   [5]   Eastlake 3rd, D., "Additional XML Security Uniform Resource
         Identifiers (URIs)", RFC 4051, April 2005.

   [6]   Klyne, G. and C. Newman, "Date and Time on the Internet:
         Timestamps", RFC 3339, July 2002.

   [7]   Maloney, M., Beech, D., Mendelsohn, N., and H. Thompson, "XML
         Schema Part 1: Structures", W3C REC REC-xmlschema-1-20010502,
         May 2001.

   [8]   Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes", W3C
         REC REC-xmlschema-2-20010502, May 2001.

   [9]   Eastlake, D., Boyer, J., and J. Reagle, "Exclusive XML
         Canonicalization Version 1.0", W3C REC REC-xml-exc-c14n-
         20020718, July 2002.

   [10] International Telecommunications Union, "Information technology
         - Open Systems Interconnection - The Directory: Public-key and
         attribute certificate frameworks", ITU-T Recommendation X.509,
         ISO Standard 9594-8, March 2000.

   [11] International Organization for Standardization, "Codes for the
         representation of names of countries and their subdivisions --
         Part 1: Country codes, 2nd edition", ISO Standard 3166,
         November 2006.

   [12] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
         January 2004.

11.2. Informative References

   [13] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", RFC
         4930, May 2007.

   [14] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms
         and Identifiers for RSA Cryptography for use in the Internet
         X.509 Public Key Infrastructure Certificate and Certificate
         Revocation List (CRL) Profile", RFC 4055, June 2005.

   [15]   Mayrhofer, A. and B. Hoeneisen, "ENUM Validation
         Architecture", RFC 4725, November 2006.

   [16] Hoeneisen, B., "ENUM Validation Information Mapping for the
         Extensible Provisioning Protocol", RFC 5076, December 2007.

   [17] ITU-T, "Computerized Directory Assistance Version 2",
         Recommendation E.115v2, October 2005.

Author's Address

   Otmar Lendl
   enum.at GmbH
   Karlsplatz 1/2/9
   Wien A-1010
   Austria

   Phone: +43 1 5056416 33
   EMail: otmar.lendl@enum.at
   URI: http://www.enum.at/