Internet Engineering Task Force (IETF)
Request for Comments: 5910
Obsoletes: 4310
Category: Standards Track
ISSN: 2070-1721

J. Gould
S. Hollenbeck
VeriSign, Inc.

May 2010

Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for the provisioning and management of Domain Name System security (DNSSEC) extensions for domain names stored in a shared central repository. Specified in XML, this mapping extends the EPP domain name mapping to provide additional features required for the provisioning of DNS security extensions. This document obsoletes RFC 4310.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5910.

Table of Contents

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) described in RFC 5730 [RFC5730]. This mapping, an extension of the domain name mapping described in RFC 5731 [RFC5731], is specified using the Extensible Markup Language (XML) 1.0 [W3C.REC-xml-20001006] and XML Schema notation ([W3C.REC-xmlschema-1-20010502] [W3C.REC-xmlschema-2-20010502]).

The EPP core protocol specification [RFC5730] provides a complete description of EPP command and response structures. A thorough understanding of the base protocol specification is necessary to understand the mapping described in this document. Familiarity with the Domain Name System (DNS) described in RFC 1034 [RFC1034] and RFC 1035 [RFC1035] and with DNS security extensions described in RFC 4033 [RFC4033], RFC 4034 [RFC4034], and RFC 4035 [RFC4035] is required to understand the DNS security concepts described in this document.

The EPP mapping described in this document specifies a mechanism for the provisioning and management of DNS security extensions in a shared central repository. Information exchanged via this mapping can be extracted from the repository and used to publish DNSSEC Delegation Signer (DS) resource records (RRs) as described in RFC 4034 [RFC4034].

This document obsoletes RFC 4310 [RFC4310]; thus, secDNS-1.1 as defined in this document deprecates secDNS-1.0 [RFC4310]. The motivation behind obsoleting RFC 4310 [RFC4310] includes:

- Addressing the issue with removing DS data based on the non-unique <secDNS:keyTag> element. The client should explicitly specify the DS data to be removed, by using all four <secDNS:dsData> elements that are guaranteed to be unique.

- Adding the ability to add and remove <secDNS:dsData> elements in a single command. This makes it consistent with RFC 5731 [RFC5731].

- Clarifying and correcting the usage of the <secDNS:chg> element. RFC 4310 [RFC4310] defined the <secDNS:chg> element as a replacement for the DS data. This is inconsistent with RFC 5731 [RFC5731], where a <domain:chg> element is used to change the values of the domain attributes.

- Adding support for the Key Data Interface described in Section 4.2 for "thick" DNSSEC servers that accept only key data and generate the associated DS data.

1.1. Conventions Used in This Document

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 BCP 14, RFC 2119 [RFC2119].

In examples, "C:" represents lines sent by a protocol client, and "S:" represents lines returned by a protocol server. "////" is used to note element values that have been shortened to better fit page boundaries. Indentation and white space in examples is provided only to illustrate element relationships and is not a mandatory feature of this protocol.

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

secDNS-1.0 is used as an abbreviation for urn:ietf:params:xml:ns:secDNS-1.0, and secDNS-1.1 is used as an abbreviation for urn:ietf:params:xml:ns:secDNS-1.1.

2. Migrating from RFC 4310

This section includes implementation recommendations for clients and servers to use in migrating from secDNS-1.0 [RFC4310] to secDNS-1.1.

As this document deprecates RFC 4310 [RFC4310], if a server announces support for both secDNS-1.0 [RFC4310] and secDNS-1.1 in the EPP greeting, clients supporting both versions SHOULD prefer secDNS-1.1.

A server SHOULD do the following to help clients migrate from secDNS-1.0 [RFC4310] to secDNS-1.1 as defined in this document.

1. A server migrating from secDNS-1.0 [RFC4310] to secDNS-1.1 SHOULD support both versions (i.e., secDNS-1.0 and secDNS-1.1) for a reasonable migration period.

2. The version of the <secDNS:infData> element to be returned by the server in the response to a <domain:info> response SHOULD depend on the <extURI> elements (indicating the secDNS extension) the client included in the EPP <login> command using the following mapping:

   - Return version secDNS-1.1 of the <secDNS:infData> element if urn:ietf:params:xml:ns:secDNS-1.1 was included as an <extURI> element in the EPP <login> command, independent of whether

urn:ietf:params:xml:ns:secDNS-1.0 is also included as an <extURI> element in the EPP <login> command.

- Return version secDNS-1.0 of the <secDNS:infData> element if urn:ietf:params:xml:ns:secDNS-1.0 but not urn:ietf:params:xml:ns:secDNS-1.1 was included as an <extURI> element in the EPP <login> command.

- Don't return the <secDNS:infData> element if neither urn:ietf:params:xml:ns:secDNS-1.0 nor urn:ietf:params:xml:ns:secDNS-1.1 was included as an <extURI> element in the EPP <login> command.

3. Object Attributes

This extension adds additional elements to the EPP domain name mapping [RFC5731]. Only those new elements are described here.

3.1. Delegation Signer Information

Delegation Signer (DS) information is published by a DNS server to indicate that a child zone is digitally signed and that the parent zone recognizes the indicated key as a valid zone key for the child zone. A DS resource record (RR) contains four fields: a key tag field, a key algorithm number octet, an octet identifying a digest algorithm, and a digest field. See RFC 4034 [RFC4034] for specific field formats.

3.1.1. Public Key Information

Public key information provided by a client maps to the DNSKEY RR presentation field formats described in Section 2.2 of RFC 4034 [RFC4034]. A DNSKEY RR contains four fields: flags, a protocol octet, an algorithm number octet, and a public key.

3.2. Booleans

Boolean values MUST be represented in the XML Schema format described in Part 2 of the W3C XML Schema recommendation [W3C.REC-xmlschema-2-20010502].

3.3. Maximum Signature Lifetime

Maximum signature lifetime (maxSigLife) is an OPTIONAL child preference for the number of seconds after signature generation when the parent's signature on the DS information provided by the child will expire. The maxSigLife value applies to the RRSIG resource

record (RR) over the DS RRset. See Section 3 of RFC 4034 [RFC4034] for information on the RRSIG resource record (RR).

The maximum signature lifetime is represented using the <secDNS: maxSigLife> element. The maxSigLife value MUST be represented in seconds, using an extended XML Schema "int" format. The base "int" format, which allows negative numbers, is described in Part 2 of the W3C XML Schema recommendation [W3C.REC-xmlschema-2-20010502]. This format is further restricted to enforce a minimum value of 1.

If maxSigLife is not provided by the client, or if the server does not support the client-specified maxSigLife value, the default signature expiration policy of the server operator (as determined using an out-of-band mechanism) applies.

4. DS Data Interface and Key Data Interface

This document describes operational scenarios in which a client can create, add, and remove Delegation Signer (DS) information or key data information for a domain name. There are two different forms of interfaces that a server can support. The first is called the "DS Data Interface", where the client is responsible for the creation of the DS information and is required to pass DS information when performing adds and removes. The server is required to pass DS information for <domain:info> responses. The second is the "Key Data Interface," where the client is responsible for passing the key data information when performing adds and removes. The server is responsible for passing key data information for <domain:info> responses.

The server MUST support one form of interface within a single command or response, where <secDNS:dsData> and <secDNS:keyData> MUST NOT be mixed, except for when <secDNS:keyData> is a child element of <secDNS:dsData> for server validation. The server MUST support the use of only one form of interface across all <secDNS:create>, <secDNS:update>, and <secDNS:infData> elements, except during a transition period, during which the server MAY support both. For instance, during a transition period, the server MAY support either the DS Data Interface or the Key Data Interface on a per-domain basis and allow the client to migrate to the target interface. The client can replace the interface used by utilizing the <secDNS:rem><secDNS: all>true</secDNS:all></secDNS:rem> element to remove all data of the old interface, and by utilizing the <secDNS:add> to add data using the new interface (<secDNS:dsData> for the DS Data Interface and <secDNS:keyData> for the Key Data Interface). The server MUST return an EPP error result code of 2306 if the server receives a command using an unsupported interface.

4.1. DS Data Interface

The DS Data Interface relies on the use of the <secDNS:dsData> element for creates, adds, removes, and <domain:info> responses. The key data associated with the DS information MAY be provided by the client, but the server is not obligated to use the key data. The server operator MAY also issue out-of-band DNS queries to retrieve the key data from the registered domain's apex in order to evaluate the received DS information. It is RECOMMENDED that the child zone operator have this key data online in the DNS tree to allow the parent zone administrator to validate the data as necessary. The key data SHOULD have the Secure Entry Point (SEP) bit set as described in RFC 3757 [RFC3757] and RFC 4034 [RFC4034].

The <secDNS:dsData> element contains the following child elements:

- A <secDNS:keyTag> element that contains a key tag value as described in Section 5.1.1 of RFC 4034 [RFC4034]. The <secDNS:keyTag> element is represented as an unsignedShort [W3C.REC-xmlschema-2-20010502].

- A <secDNS:alg> element that contains an algorithm value as described in Section 5.1.2 of RFC 4034 [RFC4034].

- A <secDNS:digestType> element that contains a digest type value as described in Section 5.1.3 of RFC 4034 [RFC4034].

- A <secDNS:digest> element that contains a digest value as described in Section 5.1.4 of RFC 4034 [RFC4034]. The <secDNS:digest> element is represented as a hexBinary [W3C.REC-xmlschema-2-20010502].

- An OPTIONAL <secDNS:keyData> element that describes the key data used as input in the DS hash calculation for use in server validation. The <secDNS:keyData> element contains the child elements defined in Section 4.2.

4.2. Key Data Interface

The Key Data Interface relies on the use of the <secDNS:keyData> element for creates, adds, removes, and <domain:info> responses. The DS information is not provided by the client but is generated by the server. The attributes used for DS generation are based on server policy, where only key data is passed between the client and the server.

The <secDNS:keyData> element contains the following child elements:

- A <secDNS:flags> element that contains a flags field value as described in Section 2.1.1 of RFC 4034 [RFC4034].

- A <secDNS:protocol> element that contains a protocol field value as described in Section 2.1.2 of RFC 4034 [RFC4034].

- A <secDNS:alg> element that contains an algorithm number field value as described in Section 2.1.3 of RFC 4034 [RFC4034].

- A <secDNS:pubKey> element that contains an encoded public key field value as described in Section 2.1.4 of RFC 4034 [RFC4034]. The <secDNS:pubKey> element is represented as a base64Binary [W3C.REC-xmlschema-2-20010502] with a minimum length of 1.

4.3. Example DS Data Interface and Key Data Interface

Example use of the secDNS-1.1 DS Data Interface for a create:

```
<secDNS:dsData>
  <secDNS:keyTag>12345</secDNS:keyTag>
  <secDNS:alg>3</secDNS:alg>
  <secDNS:digestType>1</secDNS:digestType>
  <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
</secDNS:dsData>
```

Example use of secDNS-1.1 DS Data Interface with option key data for a create:

```
<secDNS:dsData>
  <secDNS:keyTag>12345</secDNS:keyTag>
  <secDNS:alg>3</secDNS:alg>
  <secDNS:digestType>1</secDNS:digestType>
  <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
  <secDNS:keyData>
    <secDNS:flags>257</secDNS:flags>
    <secDNS:protocol>3</secDNS:protocol>
    <secDNS:alg>1</secDNS:alg>
    <secDNS:pubKey>AQPJ////4Q==</secDNS:pubKey>
  </secDNS:keyData>
 </secDNS:dsData>
```

Example use of the secDNS-1.1 Key Data Interface for a create:

```
 <secDNS:keyData>
   <secDNS:flags>257</secDNS:flags>
   <secDNS:protocol>3</secDNS:protocol>
   <secDNS:alg>1</secDNS:alg>
   <secDNS:pubKey>AQPJ////4Q==</secDNS:pubKey>
 </secDNS:keyData>
```

5. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for use in provisioning and managing DNS security extensions via EPP.

5.1. EPP Query Commands

EPP provides three commands to retrieve object information: <check> to determine if an object is known to the server, <info> to retrieve detailed information associated with an object, and <transfer> to retrieve object transfer status information.

5.1.1. EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in the EPP domain mapping [RFC5731].

5.1.2. EPP <info> Command

This extension does not add any elements to the EPP <info> command described in the EPP domain mapping [RFC5731]. However, additional elements are defined for the <info> response.

When an <info> command has been processed successfully, the EPP <resData> element MUST contain child elements as described in the EPP domain mapping [RFC5731]. In addition, the EPP <extension> element SHOULD contain a child <secDNS:infData> element that identifies the extension namespace if the domain object has data associated with this extension and based on server policy. The <secDNS:infData> element contains the following child elements:

- An OPTIONAL <secDNS:maxSigLife> element that indicates a child's preference for the number of seconds after signature generation when the parent's signature on the DS information provided by the child will expire. maxSigLife is described in Section 3.3.

- One or more <secDNS:dsData> elements or <secDNS:keyData> elements, but not both, as defined in Section 4. The <secDNS:dsData> elements describe the Delegation Signer (DS) data provided by the client for the domain. The <secDNS:keyData> elements describe the key data provided by the client for the domain. Child elements of the <secDNS:dsData> element are described in Section 4.1. Child elements of the <secDNS:keyData> element are described in Section 4.2.

Example <info> Response for a Secure Delegation Using the DS Data Interface:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:          <domain:hostObj>ns2.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:host>ns1.example.com</domain:host>
S:        <domain:host>ns2.example.com</domain:host>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:upID>ClientX</domain:upID>
S:        <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:        <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
S:        <domain:trDate>2000-04-08T09:00:00.0Z</domain:trDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <secDNS:infData
```

```
S:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
S:        <secDNS:dsData>
S:          <secDNS:keyTag>12345</secDNS:keyTag>
S:          <secDNS:alg>3</secDNS:alg>
S:          <secDNS:digestType>1</secDNS:digestType>
S:          <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
S:        </secDNS:dsData>
S:      </secDNS:infData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example <info> Response for a Secure Delegation Using the DS Data Interface with OPTIONAL Key Data:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:          <domain:hostObj>ns2.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:host>ns1.example.com</domain:host>
S:        <domain:host>ns2.example.com</domain:host>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:upID>ClientX</domain:upID>
S:        <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:        <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
S:        <domain:trDate>2000-04-08T09:00:00.0Z</domain:trDate>
```

```
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <secDNS:infData
S:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
S:        <secDNS:maxSigLife>604800</secDNS:maxSigLife>
S:        <secDNS:dsData>
S:          <secDNS:keyTag>12345</secDNS:keyTag>
S:          <secDNS:alg>3</secDNS:alg>
S:          <secDNS:digestType>1</secDNS:digestType>
S:          <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
S:          <secDNS:keyData>
S:            <secDNS:flags>257</secDNS:flags>
S:            <secDNS:protocol>3</secDNS:protocol>
S:            <secDNS:alg>1</secDNS:alg>
S:            <secDNS:pubKey>AQPJ////4Q==</secDNS:pubKey>
S:          </secDNS:keyData>
S:        </secDNS:dsData>
S:      </secDNS:infData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example <info> Response for a Secure Delegation Using the Key Data Interface:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
```

```
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:          <domain:hostObj>ns2.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:host>ns1.example.com</domain:host>
S:        <domain:host>ns2.example.com</domain:host>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:upID>ClientX</domain:upID>
S:        <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:        <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
S:        <domain:trDate>2000-04-08T09:00:00.0Z</domain:trDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <secDNS:infData
S:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
S:        <secDNS:keyData>
S:          <secDNS:flags>257</secDNS:flags>
S:          <secDNS:protocol>3</secDNS:protocol>
S:          <secDNS:alg>1</secDNS:alg>
S:          <secDNS:pubKey>AQPJ////4Q==</secDNS:pubKey>
S:        </secDNS:keyData>
S:      </secDNS:infData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if an <info> command cannot be processed for any reason.

5.1.3. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping [RFC5731].

5.2. EPP Transform Commands

EPP provides five commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes, and <update> to change information associated with an object.

5.2.1. EPP <create> Command

This extension defines additional elements for the EPP <create> command described in the EPP domain mapping [RFC5731]. No additional elements are defined for the EPP <create> response.

The EPP <create> command provides a transform operation that allows a client to create a domain object. In addition to the EPP command elements described in the EPP domain mapping [RFC5731], the command MUST contain an <extension> element, and the <extension> element MUST contain a child <secDNS:create> element that identifies the extension namespace if the client wants to associate data defined in this extension to the domain object. The <secDNS:create> element contains the following child elements:

- An OPTIONAL <secDNS:maxSigLife> element that indicates a child's preference for the number of seconds after signature generation when the parent's signature on the DS information provided by the child will expire. maxSigLife is described in Section 3.3. If the server does not support the <secDNS:maxSigLife> element, a 2102 error MUST be returned.

- Zero or more <secDNS:dsData> elements or <secDNS:keyData> elements, but not both, as defined in Section 4. Child elements of the <secDNS:dsData> element are described in Section 4.1. Child elements of the <secDNS:keyData> element are described in Section 4.2.

Example <create> Command for a Secure Delegation
Using the DS Data Interface:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <create>
C:      <domain:create
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:        <domain:period unit="y">2</domain:period>
C:        <domain:ns>
C:          <domain:hostObj>ns1.example.com</domain:hostObj>
C:          <domain:hostObj>ns2.example.com</domain:hostObj>
C:        </domain:ns>
C:        <domain:registrant>jd1234</domain:registrant>
C:        <domain:contact type="admin">sh8013</domain:contact>
C:        <domain:contact type="tech">sh8013</domain:contact>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:create>
C:    </create>
C:    <extension>
C:      <secDNS:create
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:maxSigLife>604800</secDNS:maxSigLife>
C:        <secDNS:dsData>
C:          <secDNS:keyTag>12345</secDNS:keyTag>
C:          <secDNS:alg>3</secDNS:alg>
C:          <secDNS:digestType>1</secDNS:digestType>
C:          <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
C:        </secDNS:dsData>
C:      </secDNS:create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <create> Command for a Secure Delegation
Using the DS Data Interface with OPTIONAL Key Data:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <create>
C:      <domain:create
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:        <domain:period unit="y">2</domain:period>
C:        <domain:ns>
C:          <domain:hostObj>ns1.example.com</domain:hostObj>
C:          <domain:hostObj>ns2.example.com</domain:hostObj>
C:        </domain:ns>
C:        <domain:registrant>jd1234</domain:registrant>
C:        <domain:contact type="admin">sh8013</domain:contact>
C:        <domain:contact type="tech">sh8013</domain:contact>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:create>
C:    </create>
C:    <extension>
C:      <secDNS:create
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:maxSigLife>604800</secDNS:maxSigLife>
C:        <secDNS:dsData>
C:          <secDNS:keyTag>12345</secDNS:keyTag>
C:          <secDNS:alg>3</secDNS:alg>
C:          <secDNS:digestType>1</secDNS:digestType>
C:          <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
C:          <secDNS:keyData>
C:            <secDNS:flags>257</secDNS:flags>
C:            <secDNS:protocol>3</secDNS:protocol>
C:            <secDNS:alg>1</secDNS:alg>
C:            <secDNS:pubKey>AQPJ////4Q==</secDNS:pubKey>
C:          </secDNS:keyData>
C:        </secDNS:dsData>
C:      </secDNS:create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <create> Command for a Secure Delegation Using the Key Data Interface:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <create>
C:      <domain:create
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:        <domain:period unit="y">2</domain:period>
C:        <domain:ns>
C:          <domain:hostObj>ns1.example.com</domain:hostObj>
C:          <domain:hostObj>ns2.example.com</domain:hostObj>
C:        </domain:ns>
C:        <domain:registrant>jd1234</domain:registrant>
C:        <domain:contact type="admin">sh8013</domain:contact>
C:        <domain:contact type="tech">sh8013</domain:contact>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:create>
C:    </create>
C:    <extension>
C:      <secDNS:create
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:keyData>
C:          <secDNS:flags>257</secDNS:flags>
C:          <secDNS:protocol>3</secDNS:protocol>
C:          <secDNS:alg>1</secDNS:alg>
C:          <secDNS:pubKey>AQPJ////4Q==</secDNS:pubKey>
C:        </secDNS:keyData>
C:      </secDNS:create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <create> command has been processed successfully, the EPP response is as described in the EPP domain mapping [RFC5731].

5.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the EPP domain mapping [RFC5731].

5.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the EPP domain mapping [RFC5731].

5.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping [RFC5731].

5.2.5. EPP <update> Command

This extension defines additional elements for the EPP <update> command described in the EPP domain mapping [RFC5731]. No additional elements are defined for the EPP <update> response.

The EPP <update> command provides a transform operation that allows a client to modify the attributes of a domain object. In addition to the EPP command elements described in the EPP domain mapping, the command MUST contain an <extension> element, and the <extension> element MUST contain a child <secDNS:update> element that identifies the extension namespace if the client wants to update the domain object with data defined in this extension. The <secDNS:update> element contains a <secDNS:add> element to add security information to a delegation, a <secDNS:rem> element to remove security information from a delegation, or a <secDNS:chg> element to change existing security information. At least one <secDNS:add>, <secDNS:rem>, or <secDNS:chg> element MUST be provided. The order of the <secDNS:rem> and <secDNS:add> elements is significant, where the server MUST first remove the existing elements prior to adding the new elements.

The <secDNS:update> element also contains an OPTIONAL "urgent" attribute that a client can use to ask the server operator to complete and implement the update request with high priority. This attribute accepts boolean values as described in Section 3.2; the default value is boolean false. "High priority" is relative to standard server operator policies that are determined using an out-of-band mechanism. A server MUST return an EPP error result code of 2102 if the "urgent" attribute is specified with a value of boolean true and the server does not support it. A server MUST return an EPP error result code of 2306 if the server supports the "urgent" attribute and an urgent update (noted with an "urgent" attribute value of boolean true) cannot be completed with high priority.

The <secDNS:update> element contains the following child elements:

- An OPTIONAL <secDNS:rem> element that contains a <secDNS:all> element, or one or more <secDNS:dsData> or <secDNS:keyData> elements that are used to remove security data from a delegation.

  The <secDNS:all> element is used to remove all DS and key data with a value of boolean true. A value of boolean false will do nothing. Removing all DS information can remove the ability of the parent to secure the delegation to the child zone.

  The <secDNS:dsData> element is part of the DS Data Interface and is used to uniquely define the DS record to be removed, by using all four elements -- <secDNS:keyTag>, <secDNS:alg>, <secDNS:digestType>, and <secDNS:digest> -- that are guaranteed to be unique.

  The <secDNS:keyData> element is part of the Key Data Interface and is used to uniquely define the key data to be removed, by using all four elements -- <secDNS:flags>, <secDNS:protocol>, <secDNS:alg>, and <secDNS:pubKey> -- that are guaranteed to be unique. There can be more than one DS record created for each key, so removing a key could remove more than one DS record.

- An OPTIONAL <secDNS:add> element that is used to add security information to an existing set. The <secDNS:add> element MUST contain one or more <secDNS:dsData> or <secDNS:keyData> elements. Child elements of the <secDNS:dsData> element are described in Section 4.1. Child elements of the <secDNS:keyData> element are described in Section 4.2.

- An OPTIONAL <secDNS:chg> element that contains security information to be changed. A <secDNS:chg> element contains the following child elements:

  - An OPTIONAL <secDNS:maxSigLife> element that indicates a child's preference for the number of seconds after signature generation when the parent's signature on the DS information provided by the child will expire. maxSigLife is described in Section 3.3. If the server does not support the <secDNS:maxSigLife> element, a 2102 error MUST be returned.

Example <update> Command, Adding and Removing DS Data Using the DS Data Interface:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <update>
C:      <domain:update
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <secDNS:update
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:rem>
C:          <secDNS:dsData>
C:            <secDNS:keyTag>12345</secDNS:keyTag>
C:            <secDNS:alg>3</secDNS:alg>
C:            <secDNS:digestType>1</secDNS:digestType>
C:            <secDNS:digest>38EC35D5B3A34B33C99B</secDNS:digest>
C:          </secDNS:dsData>
C:        </secDNS:rem>
C:        <secDNS:add>
C:          <secDNS:dsData>
C:            <secDNS:keyTag>12346</secDNS:keyTag>
C:            <secDNS:alg>3</secDNS:alg>
C:            <secDNS:digestType>1</secDNS:digestType>
C:            <secDNS:digest>38EC35D5B3A34B44C39B</secDNS:digest>
C:          </secDNS:dsData>
C:        </secDNS:add>
C:      </secDNS:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> Command,
Updating the maxSigLife:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <update>
C:      <domain:update
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <secDNS:update
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:chg>
C:          <secDNS:maxSigLife>605900</secDNS:maxSigLife>
C:        </secDNS:chg>
C:      </secDNS:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> Command, Adding and Removing Key Data Using the Key Data Interface, and Setting maxSigLife:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <update>
C:      <domain:update
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <secDNS:update
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:rem>
C:          <secDNS:keyData>
C:            <secDNS:flags>257</secDNS:flags>
C:            <secDNS:protocol>3</secDNS:protocol>
C:            <secDNS:alg>1</secDNS:alg>
C:            <secDNS:pubKey>AQPJ////4QQQ</secDNS:pubKey>
C:          </secDNS:keyData>
C:        </secDNS:rem>
C:        <secDNS:add>
C:          <secDNS:keyData>
C:            <secDNS:flags>257</secDNS:flags>
C:            <secDNS:protocol>3</secDNS:protocol>
C:            <secDNS:alg>1</secDNS:alg>
C:            <secDNS:pubKey>AQPJ////4Q==</secDNS:pubKey>
C:          </secDNS:keyData>
C:        </secDNS:add>
C:        <secDNS:chg>
C:          <secDNS:maxSigLife>605900</secDNS:maxSigLife>
C:        </secDNS:chg>
C:      </secDNS:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> Command, Removing DS Data with <secDNS:dsData> Using the DS Data Interface:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <update>
C:      <domain:update
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <secDNS:update
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:rem>
C:          <secDNS:dsData>
C:            <secDNS:keyTag>12346</secDNS:keyTag>
C:            <secDNS:alg>3</secDNS:alg>
C:            <secDNS:digestType>1</secDNS:digestType>
C:            <secDNS:digest>38EC35D5B3A34B44C39B</secDNS:digest>
C:          </secDNS:dsData>
C:        </secDNS:rem>
C:      </secDNS:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> Command,
Removing all DS and Key Data Using <secDNS:rem>
with <secDNS:all>:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <update>
C:      <domain:update
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <secDNS:update urgent="true"
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.0">
C:        <secDNS:rem>
C:          <secDNS:all>true</secDNS:all>
C:        </secDNS:rem>
C:      </secDNS:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example Urgent <update> Command,
Replacing all DS Data Using the DS Data Interface:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
C:  <command>
C:    <update>
C:      <domain:update
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <secDNS:update urgent="true"
C:       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:        <secDNS:rem>
C:          <secDNS:all>true</secDNS:all>
C:        </secDNS:rem>
C:        <secDNS:add>
C:          <secDNS:dsData>
C:            <secDNS:keyTag>12346</secDNS:keyTag>
C:            <secDNS:alg>3</secDNS:alg>
C:            <secDNS:digestType>1</secDNS:digestType>
C:            <secDNS:digest>38EC35D5B3A34B44C39B</secDNS:digest>
C:          </secDNS:dsData>
C:        </secDNS:add>
C:      </secDNS:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an extended <update> command has been processed successfully, the EPP response is as described in the EPP domain mapping [RFC5731].

6. Formal Syntax

An EPP object mapping is specified in XML Schema notation. The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. 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.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="urn:ietf:params:xml:ns:secDNS-1.1"
  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      domain name extension schema
      for provisioning DNS security (DNSSEC) extensions.
    </documentation>
  </annotation>

  <!--
  Child elements found in EPP commands.
  -->
```

```
    <element name="create" type="secDNS:dsOrKeyType"/>
    <element name="update" type="secDNS:updateType"/>

    <!--
    Child elements supporting either the
    dsData or the keyData interface.
    -->
    <complexType name="dsOrKeyType">
      <sequence>
        <element name="maxSigLife" type="secDNS:maxSigLifeType"
        minOccurs="0"/>
        <choice>
          <element name="dsData" type="secDNS:dsDataType"
          maxOccurs="unbounded"/>
          <element name="keyData" type="secDNS:keyDataType"
          maxOccurs="unbounded"/>
        </choice>
          </sequence>
    </complexType>

    <!--
    Definition for the maximum signature lifetime (maxSigLife)
    -->
    <simpleType name="maxSigLifeType">
      <restriction base="int">
        <minInclusive value="1"/>
      </restriction>
    </simpleType>

    <!--
    Child elements of dsData used for dsData interface
    -->
    <complexType name="dsDataType">
      <sequence>
        <element name="keyTag" type="unsignedShort"/>
        <element name="alg" type="unsignedByte"/>
        <element name="digestType" type="unsignedByte"/>
        <element name="digest" type="hexBinary"/>
        <element name="keyData" type="secDNS:keyDataType"
        minOccurs="0"/>
      </sequence>
    </complexType>

    <!--
    Child elements of keyData used for keyData interface
    and optionally with dsData interface
    -->
    <complexType name="keyDataType">
```

```
      <sequence>
        <element name="flags" type="unsignedShort"/>
        <element name="protocol" type="unsignedByte"/>
        <element name="alg" type="unsignedByte"/>
        <element name="pubKey" type="secDNS:keyType"/>
      </sequence>
    </complexType>

    <!--
    Definition for the public key
    -->
    <simpleType name="keyType">
      <restriction base="base64Binary">
        <minLength value="1"/>
      </restriction>
    </simpleType>

    <!--
    Child elements of the <update> element.
    -->
    <complexType name="updateType">
      <sequence>
            <element name="rem" type="secDNS:remType"
            minOccurs="0"/>
            <element name="add" type="secDNS:dsOrKeyType"
            minOccurs="0"/>
            <element name="chg" type="secDNS:chgType"
            minOccurs="0"/>
          </sequence>
      <attribute name="urgent" type="boolean" default="false"/>
    </complexType>

    <!--
    Child elements of the <rem> command.
    -->
    <complexType name="remType">
          <choice>
            <element name="all" type="boolean"/>
            <element name="dsData" type="secDNS:dsDataType"
            maxOccurs="unbounded"/>
            <element name="keyData" type="secDNS:keyDataType"
            maxOccurs="unbounded"/>
          </choice>
    </complexType>

    <!--
    Child elements supporting the <chg> element.
    -->
```

```
    <complexType name="chgType">
      <sequence>
        <element name="maxSigLife" type="secDNS:maxSigLifeType"
        minOccurs="0"/>
      </sequence>
    </complexType>

    <!--
    Child response elements.
    -->
    <element name="infData" type="secDNS:dsOrKeyType"/>

  </schema>
  END
```

7. Internationalization Considerations

EPP is represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations including UTF-8 [RFC3629]. Conformant XML processors recognize both UTF-8 and UTF-16 [RFC2781]. Though XML includes provisions to identify and use other character encodings through use of an "encoding" attribute in an <?xml?> declaration, use of UTF-8 is RECOMMENDED in environments where parser encoding support incompatibility exists.

As an extension of the EPP domain mapping [RFC5731], the internationalization requirements in the EPP domain mapping [RFC5731] are followed by this extension. This extension does not override any of the EPP domain mapping [RFC5731] internationalization features.

8. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in RFC 3688 [RFC3688]. Two URI assignments have been completed by the IANA.

Registration request for the extension namespace:

URI: urn:ietf:params:xml:ns:secDNS-1.1

Registrant Contact: IESG

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the extension XML schema:

URI: urn:ietf:params:xml:schema:secDNS-1.1

Registrant Contact: IESG

XML: See the "Formal Syntax" section of this document.

9. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP [RFC5730], the EPP domain name mapping [RFC5731], and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

As with other domain object transforms, the EPP transform operations described in this document MUST be restricted to the sponsoring client as authenticated using the mechanisms described in Sections 2.9.1.1 and 7 of RFC 5730 [RFC5730]. Any attempt to perform a transform operation on a domain object by any client other than the sponsoring client MUST be rejected with an appropriate EPP authorization error.

The provisioning service described in this document involves the exchange of information that can have an operational impact on the DNS. A trust relationship MUST exist between the EPP client and server, and provisioning of public key information MUST only be done after the identities of both parties have been confirmed using a strong authentication mechanism.

An EPP client might be acting as an agent for a zone administrator who wants to send delegation information to be signed and published by the server operator. Man-in-the-middle attacks are thus possible as a result of direct client activity or inadvertent client data manipulation.

Acceptance of a false key by a server operator can produce significant operational consequences. The child and parent zones MUST be consistent to secure the delegation properly. In the absence of consistent signatures, the delegation will not appear in the secure namespace, yielding untrustworthy query responses. If a key is compromised, a client can either remove the compromised information or update the delegation information via EPP commands using the "urgent" attribute.

Operational scenarios requiring quick removal of a secure domain delegation can be implemented using a two-step process. First, security credentials can be removed using an "urgent" update as just described. The domain can then be removed from the parent zone by changing the status of the domain to either of the EPP "clientHold" or "serverHold" domain status values. The domain can also be removed

from the zone using the EPP <delete> command, but this is a more drastic step that needs to be considered carefully before use.

Data validity checking and Delegation Signer record creation at the server require computational resources. A purposeful or inadvertent denial-of-service attack is possible if a client requests some number of update operations that exceed a server's processing capabilities. Server operators SHOULD take steps to manage command load and command processing requirements to minimize the risk of a denial-of-service attack.

The signature lifetime values provided by clients are requests that can be rejected. Blind acceptance by a server operator can have an adverse impact on a server's processing capabilities. Server operators SHOULD seriously consider adopting implementation rules to limit the range of acceptable signature lifetime values to counter potential adverse situations.

10. Acknowledgements

The authors would like to thank the following people who have provided significant contributions to the development of this document:

David Blacka, Howard Eland, Patrik Faltstrom, Olafur Gudmundsson, Bernie Hoeneisen, Ed Lewis, Klaus Malorny, Alexander Mayrhofer, Patrick Mevzek, David Smith, Andrew Sullivan, and Srikanth Veeramachaneni.

This document replaces RFC 4310 [RFC4310]. Please see the Acknowledgements section in that RFC for additional acknowledgements.

This document incorporates feedback from early implementers on the PROVREG mailing list and users.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC3757] Kolkman, O., Schlyter, J., and E. Lewis, "Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag", RFC 3757, April 2004.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, August 2009.

[W3C.REC-xml-20001006]
Maler, E., Sperberg-McQueen, C., Bray, T., and J. Paoli, "Extensible Markup Language (XML) 1.0 (Second Edition)", World Wide Web Consortium FirstEdition REC-xml-20001006, October 2000, <http://www.w3.org/TR/2000/REC-xml-20001006>.

[W3C.REC-xmlschema-1-20010502]
Beech, D., Thompson, H., Mendelsohn, N., and M. Maloney, "XML Schema Part 1: Structures", World Wide Web Consortium FirstEdition REC-xmlschema-1-20010502, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>.

[W3C.REC-xmlschema-2-20010502]
Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes", World Wide Web Consortium FirstEdition REC-xmlschema-2-20010502, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>.

11.2. Informative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2781] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4310] Hollenbeck, S., "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 4310, December 2005.

Appendix A. Changes from RFC 4310

1. Added the motivation in obsoleting RFC 4310 [RFC4310] to Section 1.

2. Updated Section 1 to add an explicit statement about deprecation of RFC 4310.

3. Added secDNS-1.0 and secDNS-1.1 abbreviation definitions in Section 1.1.

4. Updated "Data validity checking at the server..." to "Data validity checking and Delegation Signer record creation at the server..." in Section 9.

5. Added Section 2.

6. Updated the second paragraph of Section 7 to clarify that the internationalization features of [RFC5731] are followed.

7. Moved <secDNS:rem> prior to <secDNS:add> to conform to the EPP order semantics for supporting <secDNS:all> with <secDNS:rem> to remove all data, and for supporting the replace semantics previously supported by <secDNS:chg>.

8. Added support for the use of the <secDNS:all> boolean element under <secDNS:rem> to remove all DS or key data in place of using <secDNS:chg/>.

9. Updated <secDNS:add>, <secDNS:rem>, and <secDNS:chg> to function in a consistent way to the other EPP RFCs.

10. Removed support for <secDNS:rem> using just <secDNS:keyTag>.

11. Moved the <secDNS:maxSigLife> element out of the <secDNS:dsData> and <secDNS:keyData> elements and directly under the <secDNS: create> element, under the <secDNS:chg> element of the <secDNS: update> element, and under the <secDNS:infData> element. Section 3.3 element was updated to better describe the <secDNS: maxSigLife> element, and references to the <secDNS:maxSigLife> element were updated throughout the document.

12. Replaced references to urn:ietf:params:xml:schema:secDNS-1.0 with urn:ietf:params:xml:schema:secDNS-1.1, and replaced "Two URI assignments have been completed by the IANA" with "Two URI assignments have been completed by the IANA" in Section 8.

13. Added "The <secDNS:keyTag> element is represented as an unsignedShort [W3C.REC-xmlschema-2-20010502]" in Section 4.1.

14. Added "The <secDNS:digest> element is represented as a hexBinary [W3C.REC-xmlschema-2-20010502]" in Section 4.1.

15. Added "The <secDNS:pubKey> element is represented as a base64Binary [W3C.REC-xmlschema-2-20010502] with a minimum length of 1" in Section 4.2.

16. Combined "the command MUST contain an <extension> element" with the following sentence in Section 5.2.1 and Section 5.2.5.

17. Added sentence "If the server does not support the <secDNS: maxSigLife> element, a 2102 error MUST be returned" to Section 5.2.1 and Section 5.2.5.

18. Added sentence "This document replaces RFC 4310. Please see the Acknowledgements section in that RFC for additional acknowledgements" in Section 10.

19. Added "This document incorporates feedback from implementers on the PROVREG mail list and users" as well as "This document obsoletes RFC 4310" in the Abstract.

20. Removed all references to xsi:schemaLocation to be consistent with the other EPP RFCs.

21. Added the "DS Data Interface and Key Data Interface" section.

22. Moved the "create, add, remove, and replace Delegation Signer (DS) information" paragraph from the "Object Attributes" section to the "DS Data Interface" section.

23. Replaced the element descriptions in the "EPP <info> Command" section with a reference to the <secDNS:dsData> and <secDNS: keyData> elements described in the "DS Data Interface" and "Key Data Interface" sections, respectively.

24. Updated the "EPP <info> Command" section examples to include both the DS Data Interface and the Key Data Interface.

25. Updated the "EPP <create> Command" section to refer to both the use of <secDNS:dsData> and <secDNS:keyData> described in the "DS Data Interface" and "Key Data Interface" sections, respectively.

26. Updated the "EPP <create> Command" section examples to include both the DS Data Interface and the Key Data Interface.

27. Updated the "EPP <update> Command" section to describe the use of <secDNS:add>, <secDNS:rem>, and <secDNS:chg> together.

28. Updated the "EPP <update> Command" section examples to include both the DS Data Interface and the Key Data Interface. Also included additional examples of adding and removing DS data or key data.

29. Updated the "Formal Syntax" section with the updated XML schema.

30. Updated the Acknowledgements section with a new list of contributors.

31. Replaced references to RFC 3730 with references to RFC 5730.

32. Replaced references to RFC 3731 with references to RFC 5731.

33. Added clarification on when the extension MUST be included for each of the commands and responses (<secDNS:create>, <secDNS: update>, <secDNS:infData>).

34. Changed "In addition, the EPP <extension> element MUST contain a child <secDNS:infData> element" to "In addition, the EPP <extension> element SHOULD contain a child <secDNS:infData> element" and added "and based on server policy".

Authors' Addresses

James Gould
VeriSign, Inc.
21345 Ridgetop Circle
Dulles, VA 20166-6503
US

EMail: jgould@verisign.com

Scott Hollenbeck
VeriSign, Inc.
21345 Ridgetop Circle
Dulles, VA 20166-6503
US

EMail: shollenbeck@verisign.com