

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

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.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for the provisioning and management of Domain Name System security extensions (DNSSEC) 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.

Table of Contents

1. Introduction	2
1.1. Conventions Used in This Document	2
2. Object Attributes	3
2.1. Delegation Signer Information	3
2.1.1. Public Key Information	3
2.2. Booleans	3
2.3. Maximum Signature Lifetime Values	4
3. EPP Command Mapping	4
3.1. EPP Query Commands	4
3.1.1. EPP <check> Command	4
3.1.2. EPP <info> Command	4
3.1.3. EPP <transfer> Command	8
3.2. EPP Transform Commands	8
3.2.1. EPP <create> Command	8
3.2.2. EPP <delete> Command	11
3.2.3. EPP <renew> Command	11
3.2.4. EPP <transfer> Command	11

3.2.5. EPP <update> Command	11
4. Formal Syntax	15
5. Internationalization Considerations	18
6. IANA Considerations	18
7. Security Considerations	18
8. Acknowledgements	20
9. References	20
9.1. Normative References	20
9.2. Informative References	21

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) described in RFC 3730 [1]. This mapping, an extension of the domain name mapping described in RFC 3731 [2], is specified using the Extensible Markup Language (XML) 1.0 [3] and XML Schema notation ([4], [5]).

The EPP core protocol specification [1] 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 [11] and RFC 1035 [12] and with DNS security extensions described in RFC 4033 [13], RFC 4034 [6], and RFC 4035 [7] 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 as described in RFC 4034 [6].

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 [8].

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.

2. Object Attributes

This extension adds additional elements to the EPP domain name mapping [2]. Only new element descriptions are described here.

This document describes operational scenarios in which a client can create, add, remove, and replace delegation signer (DS) information. 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 [9].

2.1. 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 RR contains four fields: a key tag field, a key algorithm number octet, an octet identifying the digest algorithm used, and a digest field. See RFC 4034 [6] for specific field formats.

2.1.1. 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 [6]. A DNSKEY RR contains four fields: flags, a protocol octet, an algorithm number octet, and a public key.

2.2. Booleans

Boolean values MUST be represented in the XML Schema format described in Part 2 of the W3C XML Schema recommendation [5].

2.3. Maximum Signature Lifetime Values

Maximum signature lifetime values 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 [5]. This format is further restricted to enforce a minimum value of one.

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [1]. The command mappings described here are specifically for use in provisioning and managing DNS security extensions via EPP.

3.1. 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.

3.1.1. EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in the EPP domain mapping [2].

3.1.2. EPP <info> Command

This extension does not add any elements to the EPP <info> command described in the EPP domain mapping [2]. 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 [2]. In addition, the EPP <extension> element MUST contain a child <secDNS:infData> element that identifies the extension namespace and the location of the extension schema. The <secDNS:infData> element contains the following child elements:

One or more <secDNS:dsData> elements that describe the delegation signer data provided by the client for the domain. 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 [6].

A <secDNS:alg> element that contains an algorithm value as described in section 5.1.2 of RFC 4034 [6].

A <secDNS:digestType> element that contains a digest type value as described in section 5.1.3 of RFC 4034 [6].

A <secDNS:digest> element that contains a digest value as described in section 5.1.4 of RFC 4034 [6].

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. A client SHOULD specify the same <secDNS:maxSigLife> value for all <secDNS:dsData> elements associated with a domain. If the <secDNS:maxSigLife> is not present, or if multiple <secDNS:maxSigLife> values are requested, the default signature expiration policy of the server operator (as determined using an out-of-band mechanism) applies.

An OPTIONAL <secDNS:keyData> element that describes the key data used as input in the DS hash calculation. 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 [6].

A <secDNS:protocol> element that contains a protocol field value as described in section 2.1.2 of RFC 4034 [6].

A <secDNS:alg> element that contains an algorithm number field value as described in sections 2.1.3 of RFC 4034 [6].

A <secDNS:pubKey> element that contains an encoded public key field value as described in sections 2.1.4 of RFC 4034 [6].

Example <info> Response for a Secure Delegation:

```
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:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:    epp-1.0.xsd">
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:     xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
S:       domain-1.0.xsd">
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-1.0"
S:       xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
S:         secDNS-1.0.xsd">
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 with OPTIONAL 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:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:    epp-1.0.xsd">
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:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
S:          domain-1.0.xsd">
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-1.0"
S:          xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
S:            secDNS-1.0.xsd">
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:maxSigLife>604800</secDNS:maxSigLife>
S:      <secDNS:keyData>
S:          <secDNS:flags>256</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>
```

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

3.1.3. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping [2].

3.2. 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.

3.2.1. EPP <create> Command

This extension defines additional elements for the EPP <create> command described in the EPP domain mapping [2]. 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 [2], the command MUST contain an <extension> element. The <extension> element MUST contain a child <secDNS:create> element that identifies the extension namespace and the location of the extension schema. The <secDNS:

create> element MUST contain one or more <secDNS:dsData> elements. Child elements of the <secDNS:dsData> element are described in Section 3.1.2.

The <secDNS:dsData> element contains OPTIONAL <secDNS:maxSigLife> and <secDNS:keyData> elements. The server MUST abort command processing and respond with an appropriate EPP error if the values provided by the client can not be accepted for syntax or policy reasons.

Example <create> Command for a Secure Delegation:

```
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:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:    epp-1.0.xsd">
C:  <command>
C:    <create>
C:      <domain:create
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:          domain-1.0.xsd">
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-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
C:          secDNS-1.0.xsd">
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 with OPTIONAL 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:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:    epp-1.0.xsd">
C:  <command>
C:    <create>
C:      <domain:create
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:          domain-1.0.xsd">
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-1.0"
C:      xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
C:        secDNS-1.0.xsd">
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:maxSigLife>604800</secDNS:maxSigLife>
C:        <secDNS:keyData>
C:          <secDNS:flags>256</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>
```

When a <create> command has been processed successfully, the EPP response is as described in the EPP domain mapping [2].

3.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the EPP domain mapping [2].

3.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the EPP domain mapping [2].

3.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping [2].

3.2.5. EPP <update> Command

This extension defines additional elements for the EPP <update> command described in the EPP domain mapping [2]. 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. The <extension> element MUST contain a child <secDNS:update> element that identifies the extension namespace and the location of the extension schema. 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 replace security information with new security information.

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 2.2; the default value is boolean false. "High priority" is relative to standard server operator policies that are determined using an out-of-band mechanism.

The <secDNS:add> element is used to add DS information to an existing set. The <secDNS:add> element MUST contain one or more <secDNS:dsData> elements as described in Section 3.1.2.

The <secDNS:rem> element contains one or more <secDNS:keyTag> elements that are used to remove DS data from a delegation. The <secDNS:keyTag> element MUST contain a key tag value as described in section 5.1.1 of RFC 4034 [6]. Removing all DS information can remove the ability of the parent to secure the delegation to the child zone.

The <secDNS:chg> element is used to replace existing DS information with new DS information. The <secDNS:chg> element MUST contain one or more <secDNS:dsData> elements as described in Section 3.1.2. The data in these elements is used to replace whatever other data is currently archived for the delegation.

The <secDNS:update> element contains an OPTIONAL "urgent" attribute. In addition, the <secDNS:dsData> element contains OPTIONAL <secDNS:maxSigLife> and <secDNS:keyData> elements. The server MUST abort command processing and respond with an appropriate EPP error if the values provided by the client can not be accepted for syntax or policy reasons.

Example <update> Command, Adding DS 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:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:      epp-1.0.xsd">
C:  <command>
C:    <update>
C:      <domain:update
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:          domain-1.0.xsd">
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-1.0"
C:      xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
C:      secDNS-1.0.xsd">
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, Removing DS 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:      xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:      epp-1.0.xsd">
C:      <command>
C:      <update>
C:      <domain:update>
C:      xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:      xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:      domain-1.0.xsd">
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-1.0"
C:      xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
C:      secDNS-1.0.xsd">
C:      <secDNS:rem>
C:      <secDNS:keyTag>12345</secDNS:keyTag>
C:      </secDNS:rem>
C:      </secDNS:update>
C:      </extension>
C:      <clTRID>ABC-12345</clTRID>
C:      </command>
C: </epp>
```

Example Urgent <update> Command, Changing DS 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:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:    epp-1.0.xsd">
C:  <command>
C:    <update>
C:      <domain:update
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:          domain-1.0.xsd">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <secDNS:update urgent="1"
C:        xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
C:          secDNS-1.0.xsd">
C:        <secDNS:chg>
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:chg>
C:      </secDNS:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> Command, Changing Data to Include OPTIONAL 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:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:    epp-1.0.xsd">
C:  <command>
C:    <update>
C:      <domain:update
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:          domain-1.0.xsd">
```

```
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-1.0"
C:      xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0
C:      secDNS-1.0.xsd">
C:      <secDNS:chg>
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:maxSigLife>604800</secDNS:maxSigLife>
C:      <secDNS:keyData>
C:      <secDNS:flags>256</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:chg>
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 [2]. A server operator MUST return an EPP error result code of 2306 if an urgent update (noted with an "urgent" attribute value of boolean true) can not be completed with high priority.

4. 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-1.0"
  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.0"
  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:dsType"/>
  <element name="update" type="secDNS:updateType"/>

  <!--
  Child elements of the <create> command.
  -->
  <complexType name="dsType">
    <sequence>
      <element name="dsData" type="secDNS:dsDataType"
        maxOccurs="unbounded"/>
    </sequence>
  </complexType>

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

  <simpleType name="maxSigLifeType">
    <restriction base="int">
      <minInclusive value="1"/>
    </restriction>
  </simpleType>
</schema>
```



```
    </restriction>
  </simpleType>

  <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>

  <simpleType name="keyType">
    <restriction base="base64Binary">
      <minLength value="1"/>
    </restriction>
  </simpleType>

<!--
Child elements of the <update> command.
-->
  <complexType name="updateType">
    <choice>
      <element name="add" type="secDNS:dsType"/>
      <element name="chg" type="secDNS:dsType"/>
      <element name="rem" type="secDNS:remType"/>
    </choice>
    <attribute name="urgent" type="boolean" default="false"/>
  </complexType>

  <complexType name="remType">
    <sequence>
      <element name="keyTag" type="unsignedShort"
        maxOccurs="unbounded"/>
    </sequence>
  </complexType>

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

<!--
End of schema.
-->
</schema>
END
```

5. 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 [14]. Conformant XML processors recognize both UTF-8 and UTF-16 [15]. 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 [2], the elements, element content, attributes, and attribute values described in this document MUST inherit the internationalization conventions used to represent higher-layer domain and core protocol structures present in an XML instance that includes this extension.

6. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in RFC 3688 [10]. Two URI assignments have been completed by the IANA.

Registration request for the extension namespace:

URI: urn:ietf:params:xml:ns:secDNS-1.0

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-1.0

Registrant Contact: IESG

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

7. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP [1], the EPP domain name mapping [2], 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 3730 [1]. 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 name space, 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 at the server requires 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.

8. Acknowledgements

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

David Blacka, Olafur Gudmundsson, Mark Kosters, Ed Lewis, Dan Massey, Marcos Sanz, Sam Weiler, and Ning Zhang.

9. References

9.1. Normative References

- [1] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", RFC 3730, March 2004.
- [2] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", RFC 3731, March 2004.
- [3] Paoli, J., Sperberg-McQueen, C., Bray, T., and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C FirstEdition REC-xml-20001006, October 2000.
- [4] Maloney, M., Beech, D., Mendelsohn, N., and H. Thompson, "XML Schema Part 1: Structures", W3C REC REC-xmlschema-1-20010502, May 2001.
- [5] Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes", W3C REC REC-xmlschema-2-20010502, May 2001.
- [6] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.
- [7] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.
- [8] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [9] Kolkman, O., Schlyter, J., and E. Lewis, "Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag", RFC 3757, April 2004.

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

9.2. Informative References

- [11] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [12] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [13] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [14] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- [15] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000.

Author's Address

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

EMail: shollenbeck@verisign.com

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

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

