Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 23, 2014

R. Gieben
Google
M. Groeneweg
R. Ribbers
A.L.J. Verschuren
SIDN Labs

January 19, 2014

Key Relay Mapping for the Extensible Provisioning Protocol
draft-ietf-eppext-keyrelay-00

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for the purpose of relaying DNSSEC key material from one DNS operator to another, by using the administrative registration channel through the registrant, registrar and registry. The mapping introduces "<keyrelay>" as a new command in EPP.

This command will help facilitate changing the DNS operator of a domain while keeping the DNSSEC chain of trust intact.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 23, 2014.

Copyright Notice

Table of Contents

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.

The term "key material" denotes one or more DNSKEY resource records [RFC4034].

2. Introduction

Certain transactions for DNSSEC signed zones require an authenticated exchange of DNSSEC key material between DNS operators. Often there is no direct secure channel between the two or it is non-scalable.

One of such transactions is changing the DNS operator for DNSSEC signed zones ([I-D.koch-dnsop-dnssec-operator-change]. We suggest DNS operators use the administrative channel that is used to bootstrap the delegation to relay the key material for the zone. In this document we define a protocol extension for use in EPP that helps to implement and automate this transaction. This protocol extension introduces a new command called "<keyrelay>".

3. Relaying Key Material

The "<keyrelay>" command uses the existing authenticated EPP channel with the registry. Registrars can securely talk to the registry and as such the registry can serve as a drop box for relaying key material between them (see Figure 1).

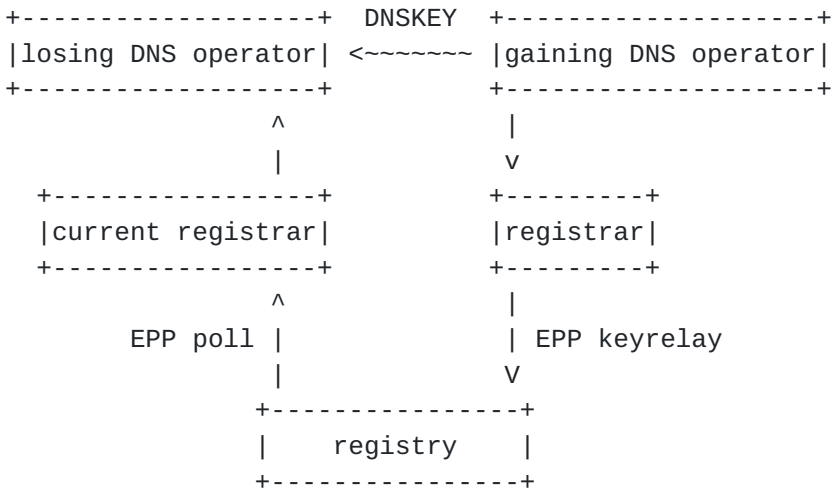

Figure 1: The gaining and losing DNS operators should talk directly to each other (the ~ arrow) to exchange the DNSKEY, but often there is no trusted path between the two. As both can securely interact with the registry over the administrative channel through the

registrar, the registry can act as a relay for the key material exchange.

The "<keyrelay>" command uploads new key(s) to the registry for a given domain. This key material is then relayed to the current registrar's message queue. This may be the same registrar as the one that submitted the "<keyrelay>" command in the situation where the DNS operators change, but the registrar stays the same. There is no need for the registry to store the relayed key in the registry system, although the registry MAY save the "<keyrelay>" message for administrative purposes.

The registrar may upload multiple keys in one "<keyrelay>" message.

There is no restriction on the type (for instance Key Signing Keys or Zone Signing Keys) of keys that can be put in the message. It is up to the gaining DNS operator to select the keys that are needed in the losing operator's zone for the intended transaction to complete successfully. It is up to the losing DNS operator to validate the correctness of the key material, and remove duplicate keys (Flags Field, Protocol Field, Algorithm Field and Public Key Field are equal) when identical keys are already in the zone.

If for some reason the registry can not process the "<keyrelay>" command an EPP error response MUST be returned. If the registry does process the "<keyrelay>" command it MUST put all uploaded keys on to the current registrar's message queue.

4. Rational For a New Command

The existing commands in EPP all deal with data which either has an owner, or soon will have one (EPP create). The "<keyrelay>" command is different, because it allows an upload of key material which will never have an owner (in the registry system). All the "<keyrelay>" command does is relay data in preparation for one of the other existing EPP commands in a process. This way, existing commands don't need to change, and backward compatibility for existing commands is guaranteed. It allows the client to be flexible in timing the individual steps necessary to complete a complex process like changing the DNS operator for a zone. Creating a separate command also allows the command to be used or extended to relay key or other data for other future processes besides DNS operator changes. This new category of EPP commands can best be described as "communication command" as it only facilitates communication of data between two EPP clients without changing any objects at the registry.

5. Key Relay Interface

The Key Relay Interface uses a "<keyrelay>" element for relaying the key material. It needs a minimum of three elements: a domain name, the key(s) to upload, a token which indicates the request is authorized by the registrant, and an OPTIONAL expire element.

Thus a "<keyrelay>" element MUST contain the following child elements:

- A "<name>" element that contains the domain name for which we upload the key.

- A "<keyData>" element that contains the key material as described in [RFC5910], Section 4.2.

- An "<authInfo>" that contains an authorization token ([RFC5731], Section 3.2.4). This indicates that the registrar has authorization from the registrant to change the zone data, and a possible future transfer is authorized. The registry MAY check if the "<authInfo>" data is correct and if it does, it MUST return an EPP error response if the authorization token is not correct.

And MAY contain:

- An "<expiry>" element that describes the expected lifetime of the relayed key(s) in the zone. The losing DNS operator can use this as an indication when to safely remove the inserted key material from the zone. This may be because the transaction that needed the insertion is either completed or has been abandoned if not completed before this expire time. The <expiry> element MUST contain one of the following child elements:

  * "<absolute/>": The policy is valid from the current date and time until it expires on the specified date and time.

  * "<relative/>": The policy is valid from the current date and time until the end of the specified duration.

  The current date and time are taken from the "<keyrelay>" service message's "<qDate>" element, see Section 7.1.

- An "<clTRID>" (client transaction identifier) as described in [RFC5730], Section 2.5.

5.1. Example Key Relay Interface

The following is an example of the "<keyrelay>" command, where a key is uploaded with a relative expire date of one month and 13 days.

```
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
C:  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:  xmlns:ext="urn:ietf:params:xml:ns:keyrelay-1.0">
C:  <extension>
C:    <ext:command>
C:      <ext:keyrelay>
C:        <ext:name>example.org</ext:name>
C:        <ext:keyData>
C:           <secDNS:flags>256</secDNS:flags>
C:           <secDNS:protocol>3</secDNS:protocol>
C:           <secDNS:alg>8</secDNS:alg>
C:           <secDNS:pubKey>cmlraXN0aGViZXN0</secDNS:pubKey>
C:        </ext:keyData>
C:        <ext:authInfo>
C:           <domain:pw>JnSdBAZSxxzJ</domain:pw>
C:        </ext:authInfo>
C:        <ext:expiry>
C:           <ext:relative>P1M13D</ext:relative>
C:        </ext:expiry>
C:      </ext:keyrelay>
C:      <ext:clTRID>ABC-12345</ext:clTRID>
C:    </ext:command>
C:  </extension>
C:</epp>
```

6. Server Reply

Example "<keyrelay>" response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:       <msg>Command completed successfully</msg>
S:    </result>
S:    <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54321-ZYX</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

As stated an EPP error response MUST be returned if a "<keyrelay>" command can not be processed for any reason.

7. Message Queue Interface

The message queue interface uses the interface as defined in [RFC5730], Section 2.6. All elements that are present in the "<keyrelay>" EPP message are put on the message queue of the current registrar for the domain in the "<name>" element.

A "<keyrelay>" message MUST be delivered to the current registrar's message queue, even if the current registrar has indicated that it does not support "<keyrelay">.

7.1. Message Queue Format

This is an example "<keyrelay>" service message. Note that the optional clTRID in this message is the clTRID value from the command that polls the message queue. It is not the clTRID value used in the EPP "<keyrelay>" command.

```
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
S:  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:  xmlns:keyrelay="urn:ietf:params:xml:ns:keyrelay-1.0">
S:  <response>
S:     <result code="1301">
S:        <msg>Command completed successfully; ack to dequeue</msg>
S:     </result>
S:     <msgQ count="5" id="12345">
S:        <qDate>1999-04-04T22:01:00.0Z</qDate>
S:        <msg>Key Relay action completed successfully.</msg>
S:     </msgQ>
S:     <resData>
```

```
S:        <keyrelay:response>
S:           <keyrelay:panData>
S:              <keyrelay:name paResult="true">example.org
S:              </keyrelay:name>
S:              <keyrelay:paDate>1999-04-04T22:01:00.0Z
S:              </keyrelay:paDate>
S:              <keyrelay:keyData>
S:                 <secDNS:flags>256</secDNS:flags>
S:                 <secDNS:protocol>3</secDNS:protocol>
S:                 <secDNS:alg>8</secDNS:alg>
S:                 <secDNS:pubKey>cmlraXN0aGViZXN0</secDNS:pubKey>
S:              </keyrelay:keyData>
S:              <keyrelay:authInfo>
S:                 <domain:pw>JnSdBAZSxxzJ</domain:pw>
S:              </keyrelay:authInfo>
S:              <keyrelay:expiry>
S:                 <keyrelay:relative>P24D</keyrelay:relative>
S:              </keyrelay:expiry>
S:              <keyrelay:reID>ClientX</keyrelay:reID>
S:              <keyrelay:acID>ClientY</keyrelay:acID>
S:           </keyrelay:panData>
S:        </keyrelay:response>
S:     </resData>
S:     <trID>
S:        <clTRID>BCD-23456</clTRID>
S:        <svTRID>65432-WXY</svTRID>
S:     </trID>
S:  </response>
S:</epp>
```

8. 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.

"<keyrelay>" command schema:

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

```
     elementFormDefault="qualified">

   <annotation>
      <documentation>
         Extensible Provisioning Protocol v1.0 domain name
         extension schema for relaying key material.
      </documentation>
   </annotation>

   <import namespace="urn:ietf:params:xml:ns:epp-1.0"
           schemaLocation="epp-1.0.xsd" />
   <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
           schemaLocation="eppcom-1.0.xsd" />
   <import namespace="urn:ietf:params:xml:ns:secDNS-1.1"
           schemaLocation="secdns-1.1.xsd" />
   <import namespace="urn:ietf:params:xml:ns:domain-1.0"
           schemaLocation="domain-1.0.xsd" />

   <element name="command" type="keyrelay:commandType" />
   <element name="response" type="keyrelay:responseType" />

   <complexType name="responseType">
      <sequence>
         <element name="panData"
                  type="keyrelay:panKeyRelayDataType"/>
      </sequence>
   </complexType>

   <complexType name="commandType">
      <sequence>
         <element name="keyrelay"
                  type="keyrelay:keyRelayType" />
         <element name="clTRID" type="epp:trIDStringType"
                  minOccurs="0"/>
      </sequence>
   </complexType>

   <complexType name="keyRelayExpiryType">
      <choice>
         <element name="absolute" type="dateTime" />
         <element name="relative" type="duration" />
      </choice>
   </complexType>

   <complexType name="keyRelayType">
      <sequence>
         <element name="name" type="eppcom:labelType" />
         <element name="keyData" type="secDNS:keyDataType"
```

```
                  minOccurs="1" maxOccurs="unbounded" />
            <element name="authInfo"
                     type="domain:authInfoType" />
            <element name="expiry"
               type="keyrelay:keyRelayExpiryType" minOccurs="0" />
         </sequence>
      </complexType>

      <complexType name="panKeyRelayDataType">
         <sequence>
            <element name="name" type="domain:paNameType" />
            <element name="paDate" type="dateTime" />
            <element name="keyData" type="secDNS:keyDataType"
                     minOccurs="1" maxOccurs="unbounded" />
            <element name="authInfo" type="domain:authInfoType" />
            <element name="expiry"
               type="keyrelay:keyRelayExpiryType" minOccurs="0" />
            <element name="reID" type="eppcom:clIDType"/>
            <element name="acID" type="eppcom:clIDType"/>
         </sequence>
      </complexType>
   </schema>
```

9. 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 must be completed by the IANA.

Registration request for the extension namespace:

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

Registrant Contact: IESG

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

10. Security Considerations

The "<keyrelay>" EPP extension does not allow for any object transformations.

Any registrar can use this mechanism to put key material on the message queue of another registrar, thus mounting a denial of service attack. However this can, and should be detected by the registry. A correct "<ext:authInfo>" element can be used as an indication that putting the key material on the losing registrar's message queue is authorized by the registrant of that registrar. A registry MAY set a server policy which limits or rejects "<keyrelay>" messages if it detects the mechanism is being abused.

Communication between a registrar and registry is mostly done over EPP, but communication between DNS operators, registrants or registrars often is not. If EPP is not used between these entities, relaying the key between a DNS operator and registrar should be adequately authenticated for the complete relay channel to remain secure. It's out of scope for this document to describe how to authenticate with other methods than EPP.

11. Acknowledgements

We like to thank the following individuals for their valuable input, review, constructive criticism in earlier revisions or support for the concepts described in this document:

Maarten Wullink, Marco Davids, Ed Lewis, James Mitchell, David Peal, Patrik Faltstrom, Klaus Malorny, James Gould, Patrick Mevzek and Seth Goldman.

12. References

12.1. 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.

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

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

[RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, May 2010.

12.2. Informative References

[I-D.koch-dnsop-dnssec-operator-change]
Koch, P., Sanz, M., and A. Verschuren, "Changing DNS Operators for DNSSEC signed Zones", draft-koch-dnsop-dnssec-operator-change-05 (work in progress), July 2013.

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

Appendix A. Changelog

[This section should be removed by the RFC editor before publishing]

A.1. draft-gieben-epp-keyrelay-00

1. Initial document.

A.2. draft-gieben-epp-keyrelay-01

1. Style and grammar changes;

2. Added an expire element as per suggestion by Klaus Malorny;

3. Make the authInfo element mandatory and make the registry check it as per feedback by Klaus Malorny and James Gould.

A.3. draft-gieben-epp-keyrelay-02

1. Added element to identify the relaying EPP client as suggested by Klaus Malorny;

2. Corrected XML for missing and excess clTRID as noted by Patrick Mevzek;

3. Added clarifications for the examples based on feedback by Patrick Mevzeck;

4. Reviewed the consistency of using DNS operator versus registrar after review comments by Patrick Faltstrom and Ed Lewis.

A.4. draft-gieben-epp-keyrelay-03

1. Style and grammar changes

2. Corrected acknowledgement section

3. Corrected XML for Expire element to not be mandatory but only occur once.

A.5. draft-eppext-epp-keyrelay-00

1. Added feedback from Seth Goldman and put him in the acknowledgement section.

2. IDnits formatting ajustments

Authors' Addresses

R. (Miek) Gieben
Google

Email: miek@google.com

M. Groeneweg
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: marc.groeneweg@sidn.nl
URI: https://www.sidn.nl/

Rik Ribbers
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: rik.ribbers@sidn.nl
URI: https://www.sidn.nl/

Antoin Verschuren
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: antoin.verschuren@sidn.nl
URI: https://www.sidn.nl/