eppext
Internet-Draft
Intended status: Standards Track
Expires: July 12, 2015

R. Gieben
Google
M. Groeneweg
H. Ribbers
SIDN Labs
A. Verschuren

January 8, 2015

Relay Extension for the Extensible Provisioning Protocol
draft-ietf-eppext-keyrelay-01

Abstract

This document describes a generic Extensible Provisioning Protocol (EPP) extension for the purpose of relaying data between registrars.

Furthermore, this document describes a specific implementation for relaying DNSSEC key material between DNS operators (by means of their respective registrars), to facilitate the change of DNS operator, while keeping the DNSSEC chain of trust intact.

This I-D introduces a new generic command <relay> and an element <relayData>. For the specific implementation of relaying DNSSEC key material it introduces an extension of the <relayData> with a <keyRelayData> element.

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 12, 2015.

Table of Contents

1. Introduction

There are certain transactions in the lifecycle of a domain name, that require interaction between registrars but need registration data from the registry. Since all registrars involved have a secure channel to the registry for maintaining the delegation, the registry can act as relay for such data to transfer securely and authoritative between the registrars involved.

Currently these transactions aren't supported in the Extensible Provisioning Protocol (EPP) [RFC5730]. One example of such a transaction is the exchange of DNSSEC key material to keep the DNSSEC chain of trust intact in case of a change of DNS-operator.

In this document we will define:

- A protocol extension that implements the relaying of data between registrars through the existing authenticated EPP channel. This protocol extension introduces a new EPP command called <relay> with an element <relayData>.

- An extension to the <relayData> element called <keyRelayData> that can be used for the relaying DNSSEC key material using the <relay> command.

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].

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.

In examples, "C:" represents lines sent by a protocol client, and "S:" represents lines returned by a protocol server. Indentation and white space in examples is provided only to illustrate element relationships and is not a mandatory feature of this protocol.

1.2. Relaying Data

The <relay> command uses the existing authenticated EPP channel between the registrar and the registry. Registrars can use this secure channel for relaying data to other registrars. The registry serves as an intermediary between two registrars (see Figure 1).

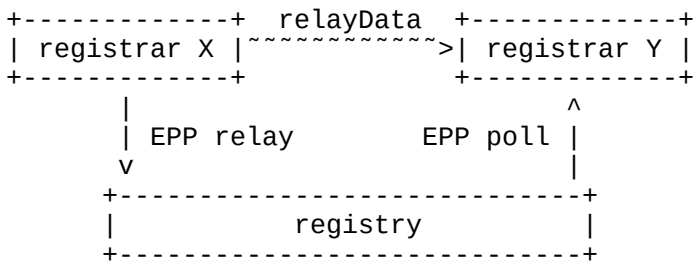

Figure 1: Registry acting as a relay for secure data exchange between registrars.

The <relay> command uploads data from a registrar X to the registry. The uploaded data is then pushed onto the message queue of registrar Y by the registry based on the information within the <relayData> element of the <relay> command and the registration data maintained by the registry.

The data to be relayed MUST relate to registration data of the registry. The <relay> command is not intended to relay data that has no relationship to registration data. We have e-mail for that.

If for some reason the registry cannot process the <relay> command, an EPP error response MUST be returned. If the registry does process the <relay> command it MUST put all elements of <relayData> on to the message queue of registrar Y.

1.2.1. Rationale For a New Command

This new <relay> command can be best described as a "transient command" as it only facilitates communication of data between two registrars without changing the registration data at the registry. No existing EPP command can be (re)used for this function. This extension of EPP is in accordance to [RFC3735].

1.2.2. Extending <relayData> per use case

One MUST extend the <relayData> element per use case to define the data to be relayed. In the extension, one MUST make provisions for the registry how to determine the receiving registrar of the <relay> command.

1.3. Secure Transfer of DNSSEC Key Material

Exchanging DNSSEC key material in preparation of a domain name transfer is one of the phases in the lifecycle of a domain name [I-D.koch-dnsop-dnssec-operator-change].

DNS-operators need to exchange (through the gaining registrar) DNSSEC key material before the registration data can be changed.

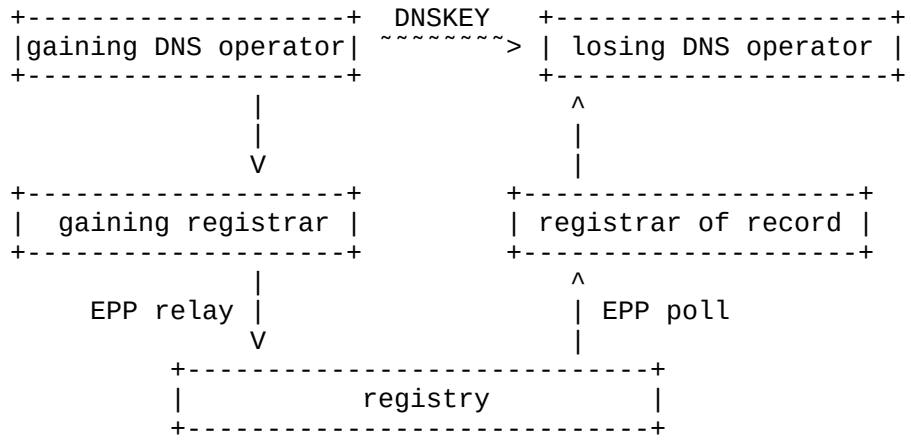

Figure 2: Transfer of DNSSEC key material.

As the <relay> command uses a secure channel, it can be used as a method for exchanging this DNSSEC key material securely (see Figure 2).

The gaining and losing DNS operators could 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.

This I-D contains an extension of the <relayData> element for this use case.

2. Object Attributes

2.1. DNSSEC Key Material

To transfer DNSSEC key material with the <relay> command the generic <relayData> is extended with a <keyRelayData> element that contains the data for relaying the key material. See Section 1.2.2.

This <keyRelayData> element REQUIRES a minimum of three child elements:

- A <name> element which contains the domain name for which we upload the key. The registry MUST relay the <keyRelayData> to the registrar of record of the provided domain name.

- One or more <keyData> elements that contains the DNSSEC key material as described in [RFC5910], Section 4.2.

- An <authInfo> element that contains an authorization token ([RFC5731], Section 3.2.1). 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 an OPTIONAL <expiry> child element.

- 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.

3. EPP Command Mapping

3.1. EPP Transient Commands

3.1.1. EPP <relay> Command

The EPP <relay> command is a generic EPP command used for relaying data between registrars. It contains the data to be relayed and the client transaction identifier. It has been designed to be extensible for usage in other use-cases.

The <relay> command REQUIRES the following child elements:

- One or more <relayData> elements containing data to be relayed.

- An OPTIONAL <clTRID> (client transaction identifier) element that MAY be used to uniquely identify the command to the registrar. See [RFC5730], Section 2.5.

Example <relay> command:

```
C:<?xml version="1.0" encoding=:UTF-8" standalone="no"?>
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:r="urn:ietf:params:xml:ns:relay-1.0">
C:    <extension>
C:        <r:relay xmlns:r="urn:ietf:params:xml:ns:relay-1.0">
C:            <r:relayData>
C:                    <k:keyRelayData
C:                       xmlns:k="urn:ietf:params:xml:ns:keyrelay-1.0">
C:                       <k:name>example.org</k:name>
C:                       <k:keyData>
C:                            <secDNS:flags>256</secDNS:flags>
C:                            <secDNS:protocol>3</secDNS:protocol>
C:                            <secDNS:alg>8</secDNS:alg>
C:                            <secDNS:pubKey>
C:                                cmlraXN0aGViZXN0</secDNS:pubKey>
C:                        </k:keyData>
C:                        <k:authInfo>
C:                            <domain:pw>JnSdBAZSxxzJ</domain:pw>
C:                        </k:authInfo>
C:                        <k:expiry>
C:                            <k:relative>P1M13D</k:relative>
C:                        </k:expiry>
C:                    </k:keyRelayData>
C:             </r:relayData>
C:            <r:clTRID>ABC-12345</r:clTRID>
C:        </r:relay>
C:    </extension>
C:</epp>
```

3.2. EPP Query Commands

This EPP extension does not change any command other than the EPP <poll> command response.

3.2.1. EPP <poll> command

This extension adds elements to the response to a <poll> command with the "op" attribute set to "req". Specifically, a <panData> element is added to the <resData> section of the service message, containing the following elements:

- A REQUIRED <relayData> element that contains the relayed data.
- A REQUIRED <paDate> element that contains the date and time of the submitted <relay> command.

- A REQUIRED <reID> element that contains the identifier of the registrar that requested the data relay.
- A REQUIRED <acID> element that contains the identifier of the registrar that SHOULD act upon the data relay.

Example <poll> response:

```
S:<?xml version="1.0" encoding=:UTF-8" standalone="no"?>
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:    <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>Relay action completed successfully.</msg>
S:        </msgQ>
S:        <resData>
S:            <r:panData xmlns:r="urn:ietf:params:xml:ns:relay-1.0">
S:                <r:relayData>
S:                    <k:keyRelayData
S:                       xmlns:k="urn:ietf:params:xml:ns:keyrelay-1.0">
S:                       <k:name>example.org</k:name>
S:                       <k:keyData>
S:                            <secDNS:flags>256</secDNS:flags>
S:                            <secDNS:protocol>3</secDNS:protocol>
S:                            <secDNS:alg>8</secDNS:alg>
S:                            <secDNS:pubKey>
S:                                cmlraXN0aGViZXN0</secDNS:pubKey>
S:                        </k:keyData>
S:                        <k:authInfo>
S:                            <domain:pw>JnSdBAZSxxzJ</domain:pw>
S:                        </k:authInfo>
S:                        <k:expiry>
S:                            <k:relative>P1M13D</k:relative>
S:                        </k:expiry>
S:                    </k:keyRelayData>
S:                </r:relayData>
S:                <r:paDate>1999-04-04T22:01:00.0Z</r:paDate>
S:                <r:reID>ClientX</r:reID>
S:                <r:acID>ClientY</r:acID>
S:            </r:panData>
S:        </resData>
S:        <trID>
S:            <clTRID>BCD-23456</clTRID>
S:            <svTRID>65432-WXY</svTRID>
S:        </trID>
S:    </response>
S:</epp>
```

4. Formal Syntax

4.1. Formal Syntax <relay> command and POLL response

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

    <annotation>
        <documentation>
            Extensible Provisioning Protocol v1.0 protocol
            extension schema for relaying data.
        </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" />

    <element name="relay" type="r:relayDataType" />
    <element name="panData" type="r:relayPanDataType" />

    <complexType name="relayDataType">
        <sequence>
            <element name="relayData" type="epp:extAnyType" />
            <element name="clTRID" type="epp:trIDStringType"
                minOccurs="0" />
        </sequence>
    </complexType>

    <complexType name="relayPanDataType">
        <sequence>
            <element name="relayData" type="epp:extAnyType" />
            <element name="paDate" type="dateTime" />
            <element name="reID" type="eppcom:clIDType" />
            <element name="acID" type="eppcom:clIDType" />
        </sequence>
    </complexType>
</schema>
```

4.2. Formal Syntax <keyRelayData> data

```
<?xml version="1.0" encoding="UTF-8"?>
    <schema targetNamespace="urn:ietf:params:xml:ns:keyrelay-1.0"
    xmlns:k="urn:ietf:params:xml:ns:keyrelay-1.0"
    xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
    xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
    xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
    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 protocol
            extension schema for relaying DNSSEC key data.
        </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="keyRelayData" type="k:keyRelayDataType" />

    <complexType name="keyRelayDataType">
        <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="k:keyRelayExpiryType"
                minOccurs="0" />
        </sequence>
    </complexType>
    <complexType name="keyRelayExpiryType">
        <choice>
            <element name="absolute" type="dateTime" />
            <element name="relative" type="duration" />
        </choice>
    </complexType>
</schema>
```

5. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in RFC3688 [RFC3688].

Four URI assignments must be completed by the IANA.

Registration request for the extension namespaces:

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

Registrant Contact: IESG

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

Registration request for the extension XML schemas:

URI: urn:ietf:params:xml:schema:keyrelay-1.0
URI: urn:ietf:params:xml:schema:relay-1.0

Registrant Contact: IESG

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

6. Security Considerations

A registry MUST NOT perform any transformation on registration data under registry management when processing a <relay> command.

Any registrar can use this mechanism to put data on the message queue of another registrar, allowing for the potential of a denial of service attack. However this can, and SHOULD be detected by the registry. A registry MAY set a server policy which limits or rejects <relay> messages if it detects the mechanism is being abused.

For the <keyRelayData> data a correct <authInfo> element SHOULD be used as an indication that putting the key material on the registrar’s message queue is authorized by the _registrant_ of that domain name. This draft does not specify how this <authInfo> is provided to the registrar. This depends on how the DNS operator is authorised to perform DNS changes on behalf of the registrant through the registrar on record. This authorisation is not covered in this I-D.

7. References

7.1. 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.

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

7.2. 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-06 (work in progress), February 2014.

[RFC3735] Hollenbeck, S., "Guidelines for Extending the Extensible Provisioning Protocol (EPP)", RFC 3735, March 2004.

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-ietf-eppext-keyrelay-00

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

2. IDnits formatting ajustments

A.6. draft-ietf-eppext-keyrelay-01

1. Introducing the <relay> command, and thus seperating the data and the command.

2. Updated the Introduction, describing the general use of relay vs the intended use-case of relaying DNSSEC key data.

3. Restructuring the document to make it more inline with exisiting EPP extensions.

Authors' Addresses

Miek Gieben
Google

Email: miek@google.com

Marc 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

Email: ietf@antoin.nl