Changing DNS Operators for DNSSEC signed Zones
draft-koch-dnsop-dnssec-operator-change-06

Abstract

   Changing the DNS delegation for a DNS zone is quite involved if done
   by the books, but most often handled pragmatically in today's
   operational practice at the top level with registries and registrars.
   This document describes a delegation change procedure that maintains
   consistency and validation under DNSSEC.

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 August 18, 2014.

Copyright Notice

Table of Contents

1. Introduction

When the NS RRSet in a DNS delegation is to be changed from one to a
disjoint other, the most conservative approach would be to add the
new servers as (stealth) secondary servers, then add them to the NS
RRSet, change the primary master (that is, change the AXFR/IXFR
source for all the secondary servers), remove the old name servers'
names from the NS RRSet and finally cease service on those former
name servers. This would involve two changes to the zone file for
the apex NS RRSet and in turn two interactions with the parent zone
(the registry) for the delegation NS RRSet. Another procedure would
at least see the old name servers acquire the new data by transfer
and then publish it until the NS records' time to live (TTL) values
expire.

Operational practice deviates from this in many cases, especially
where there is a combined role of the registrar as registrar, DNS

operator and web hoster. Often the new infrastructure is set up
independent of and in parallel to the old one and there is only one
delegation change. Resolvers will access either version of the DNS
zone and the inconsistency in DNS data and even at the application
layer will be tolerated as long as the end user gets to see any
result at all.

DNSSEC [RFC4033][RFC4034][RFC4035] is less tolerant to this
inconsistency, and the challenge is that access to and validation of
DNS data will involve multiple steps. The resolver might access DNS
data through one (say, the old) name server infrastructure and DNS
key material (the apex DNSKEY RRSet) through the other. A hard
switch would increase the likelyhood of a validation failure, should
the signature over some RRSet not match any key in the DNSKEY RRSet.

1.1. Purpose of this Document

This document attempts to list requirements for a seamless change of
DNS operators in a registry/registrar environment. It then suggests
a procedure that should work in an automated environment even for
large numbers of DNS operator changes. It is meant as a supplement
or contribution to the updated version of [RFC4641], as currently
expressed in [RFC6781], section 4.3.5.

1.2. Terminology

The change of DNS infrastructure involves multiple parties. We are
using the following terms throughout the document.

Registrar. The entity that can change entries in a DNS registry
   database, usually, but not restricted to, by means of a realtime
   provisioning protocol like EPP [RFC5730]. NB: the procedure
   described in this document does not require a strict registry-
   registrar-registrant separation. Where the registrant can
   directly interact with the registry they are considered filling
   the registrar role.

DNS operator. The DNS operator provides the DNS infrastructure for a
   given DNS zone (delegated domain). This party may or may not be
   identical to the registrant or the registrar. The details of
   provisioning the DNS data into the DNS zone are beyond the scope
   of this document.

losing DNS operator is the party that controls zone content and DNS
   infrastructure at the beginning of a DNS operator change.

   gaining DNS operator denotes the party that will control zone
      content and DNS infrastructure after the successful DNS operator
      change.

   The terms Zone Signing Key (ZSK) and Key Signing Key (KSK) are
   defined in section 2 of [RFC4033].

   Domain names and IP addresses herein are for explanatory purposes
   only and should not be expected to lead to useful information in real
   life [RFC2606],[RFC5735].

2. Requirements and Assumptions for Seamless Operator Change

   Regardless of the the particular registry provisioning protocol
   (e.g., EPP, [RFC5730], [RFC5731]) DNS registries usually provide for
   a method to transfer a domain name between different registrars.
   However, from this angle there is no separate role for the DNS
   operator, the entity that is responsible for the DNS infrastructure
   and/or the content of the DNS zone. This entity may be identical to
   the registrar, the registrant, or neither. From the DNSSEC
   perspective it is important to consider the entity controlling the
   ZSK and KSK in any transfer that involves changing the NS RRSet to a
   disjoint set (as opposed to simple additions to or removals from the
   NS RRSet).

   The change of a registrar is of limited effect from the DNSSEC
   perspective. The discussion of the ability of the gaining registrar
   to accept DNSSEC information within a domain object is beyond the
   scope of this document. The focus is kept on the change of the DNS
   operator. There are two cases: ideally, the losing operator will be
   able to incorporate data generated by the gaining operator into their
   version of the DNS zone. However, the proposed solution should also
   be able to cover the case where this cooperation cannot be relied
   upon.

2.1. Requirements for Seamless Operator Change

   This is a list of requirements for the operator change:

   no validation failures During the operator change, as in normal
      operations, DNSSEC validation failures, resulting from unavailable
      or outdated key or signature material, must be avoided.

   validation failures worse than insecure Where there is the choice
      between DNSSEC validation failures and an insecure state, the
      latter is to be preferred, although its extent still ought to be
      kept to a necessary minimum.

no private keys Private cryptographic keys will not be exchanged
   between the losing and the gaining operator. Scenarios in which
   this would be feasible would not pose the challenges addressed in
   this document.

no use of foreign RRSIG information The process of signing a zone is
   usually done in one of three ways: The zone is signed completely
   as a file, then loaded into a name server, the signer is a "bump
   in the wire" solution or the name server combines signing and
   serving on the primary master. Adding foreign signatures into a
   zone is not easily achieved in any of these setups, so it ought to
   be avoided. It also helps contain the effect of changes to an
   RRSet locally, as no signatures have to be generated remotely with
   subsequent re-import.

little interaction To ease automation the number of interactions
   between registry, registrar or registrars and operators should be
   minimized.

little overhead for losing operator Whenever interaction or action
   cannot be avoided, the losing operator should be involved as
   little as possible to avoid overhead for the leaving customer. In
   turn, this suggests the gaining operator should be in control of
   the process.

2.2. Assumptions for Seamless Operator Change

This is a list of assumptions for the operator change:

no direct communication channel between operators We do not assume
   the existence of a direct, confidential, authenticated
   communication channel between the losing and the gaining operator.
   On an abstract level, the registrant could be viewed as an
   indirect channel, but involving the registrant is not necessarily
   easily automated.

secure communication channel between operator, registrar, and registry
   We do, though, assume the existence of a direct, confidential,
   authenticated communication channel between an operator and their
   registrar as well as between the registrar and the registry.

incorporation of foreign DNSKEY RRs To achieve a symmetric
   validation chain along the DS/KSK/ZSK of both the losing and the
   gaining DNS operator it is assumed that either operator can,
   technically, incorporate DNSKEY RRs into their apex DNSKEY RRSet.
   This includes the ability to subsequently properly sign the DNSKEY
   RRSet.

ability to store and retrieve DNSKEY RRs We assume that registrars
   are able to store in and retrieve from the registry DNSKEY RRs or
   equivalent information. This can be achieved by the registry
   using DNSKEY RRs as key material (as opposed to or in addition to
   DS RRs) or by the ability to insert DNSKEY data in free text or
   remarks fields within or associated with the domain object.

no algorithm rollover Current reading is that an algorithm rollover
   requires a full validation with all algorithms involved, whereas a
   key rollover will work whenever data can be validated using either
   key ([RFC4035], section 2.2). Therefore, it is assumed that both
   operators utilize the same DNSSEC key algorithm during the
   transfer. This does neither preclude the use of different key
   sizes nor the change of key algorithms after the DNS operator
   change.

ZSK/KSK separation This document assumes a key separation between
   Zone Signing Key (ZSK) and Key Signing Key (KSK) as per [RFC4641].
   Where a zone maintainer decides to use only a single key [RFC6781]
   the process layout will remain the same, details of differences to
   be explored in a future update of this draft.

no ZSK/KSK rollover in progress Since no private keys will change
   hands, the operator change implies a key rollover from the losing
   to the gaining operator. A progressing rollover of the ZSK or KSK
   at the losing DNS operator would add complexity due to more
   possible validation paths. While both a rollover and a DNS
   operator change can be combined, we will, for the sake of
   simplicty, assume they will not. How to implement this business
   constraint is beyond the scope of this document. Since the DNS
   zone at the gaining DNS operator will be set up from scratch, it
   is assumed there is no rollover in progress, either.

resolver does not indefinitely prolongate retention of cached data
   Some resolvers are known to refresh the TTL of an NS RRSet of a
   zone upon every authoritative response they receive that carries
   this NS RRSet in the authority section. This is partly in the
   spirit of [RFC2181] in that this source (the authoritative child)
   is more credible than the NS RRSet in a referral, but independent
   of DNSSEC this behaviour leads to an undesired side effect. These
   'sticky' resolvers will never learn of a redelegation if that
   happens by switching between two non-overlapping sets of name
   servers, which is current practice in many environments.
   Therefore the process for operator change does not have to take
   this behaviour into account but may assume benign resolver
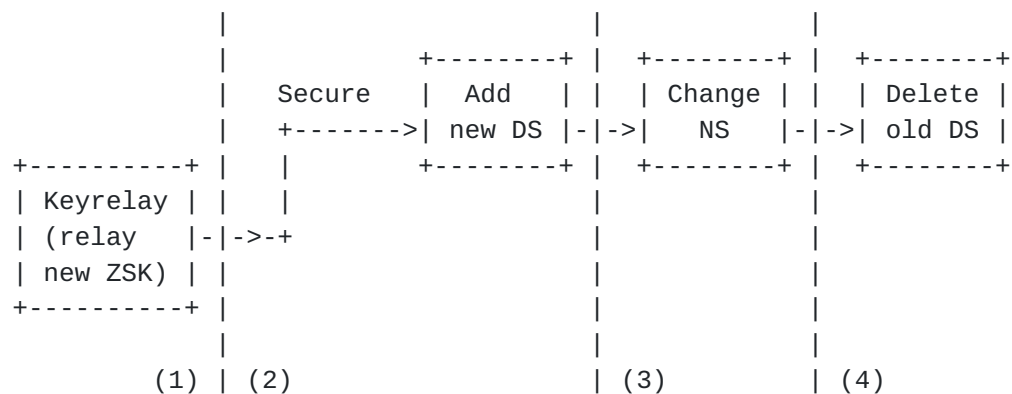   operation.

3. **Executing the Change**

   The challenge to face is that during the DNS operator change
   resolvers may see an NS RRSet pointing to either the losing or the
   gaining operator's name servers. They may also have cached DNS data
   and corresponding signatures from either source and this may in
   particular mean that the apex DNSKEY RRSet and its signature or
   signatures have a different origin than some other to-be-validated
   RRSet.

   To prevent validation failures, resolvers need, through careful
   timing, be given the chance to acquire the necessary DNSKEY data that
   allows them to validate signed DNS data from either source.
   Therefore the DNSKEY RRSets originating from either infrastructure or
   a limited time window need to have a non zero intersection set that
   will be covered by both the old an the new trust anchor (or KSK).

   Changing the DNS operator will be a three step process involving both
   the gaining and the losing DNS operator. For the sake of simplicity,
   the first case will not cover a registrar change.

3.1. **Changing DNS operator only**

   At the beginning, there is a delegation to the losing operator's name
   servers and a DS RR pointing to a KSK controlled by that losing
   registrar in the parent zone. As a first step, the gaining operator
   will obtain through the DNS the ZSK(!) from the losing operator's
   version of the zone, validate it using the publicly available DNS
   information, add it to its apex DNSKEY RRSet and re-sign that RRSet.
   At this point, the new infrastructure will not yet be queried, but it
   would provide a validation path through the new KSK for signatures
   generated with either the new or old ZSK. To achieve symmetry, the
   new ZSK needs to be incorporated into the old apex DNSKEY RRSet.
   Since a (trusted, validatable) path to that information is not
   available in the DNS, yet and no direct communication path between
   the losing and the gaining operator is assumed, the registry will act
   as a dropbox. Again, for the sake of simplicity, we assume that the
   registry accepts key material in the form DNSKEY rather than DS
   resource records.

```
                |                 |              |
                |        +--------+ |  +--------+ |  +--------+
                |        | Secure | | Add  | |  | Change | |  | Delete |
                |        | +------->| new DS |-|->|   NS   |-|->| old DS |
   +----------+ |        |        +--------+ |  +--------+ |  +--------+
   | Keyrelay | |        |                    |              |
   | (relay   |-|->-+                         |              |
   | new ZSK) | |                             |              |
   +----------+ |                             |              |
                |                             |              |
          (1) | (2)                          | (3)          | (4)
```

   (1): Gaining operator configures zone with old ZSK included and
        signs
   (2): Loosing operator recieves new ZSK, includes ZSK in zone and
        resigns
   (3): Wait max TTL old DNSKEY RRset after seeing new ZSK in old
        zone and DS TTL
   (4): Wait TTL NS RRset old zone after NS change

    Figure: Changing DNS operator only

   The first change to the registration data, to be initiated by the
   gaining operator will add the new KSK to the domain object to be
   added to the parent zone as DS. This way, the parent zone can
   publish two (or more) DS RRs, one for either KSK. At the same time,
   the registry can make the new ZSK available to the losing operator.
   No change is initiated yet to the parent zone NS RRSet.

   Upon appearance of the gaining operator's ZSK in the registry's
   communication channel to the losing operator, he is supposed to copy
   this ZSK into its version of the apex DNSKEY RRSet and to re-sign the
   DNSKEY RRSet with the old KSK.

   Once that has happened and the TTL value of the (previous, sans new
   ZSK) DNSKEY RRSet has passed, all resolver caches will either have no
   DNSKEY RRSet for this zone or they will have acquired one that has a
   signature made by the old KSK over both the old and the new ZSK. The
   second precondition for the next step is that no DS RRSets exist
   without reference to the new KSK that was inserted in the previous,
   first step. This will be the case after the new DS RRSet will have
   been visible in the DNS and the TTL of the DS RRSet (actually, that
   of the previous instance of that DS RRSet) will have expired. Both
   expiration intervals may and likely will overlap, but may start at
   different times and are otherwise independent. The termination of
   the later of the two will determine the earliest point in time for
   progress.

The gaining operator can now initiate the second step, changing the
NS RRSet in the parent to the new name server infrastructure. Note
that a hard delegation change rather than a multi-step phase-in is
part of the design to reflect today's operational practice.

After this second step, the parent zone will contain an NS RRSet
delegating to the gaining operator's name servers as well as two DS
records, one referring to the old KSK, the other referring to the new
KSK. Due to the two DS RRs, either DNSKEY RRSet can be validated.
Either path will provide validation for both the old and the new ZSK.
That enables RRSIG validation on all DNS data independent of its
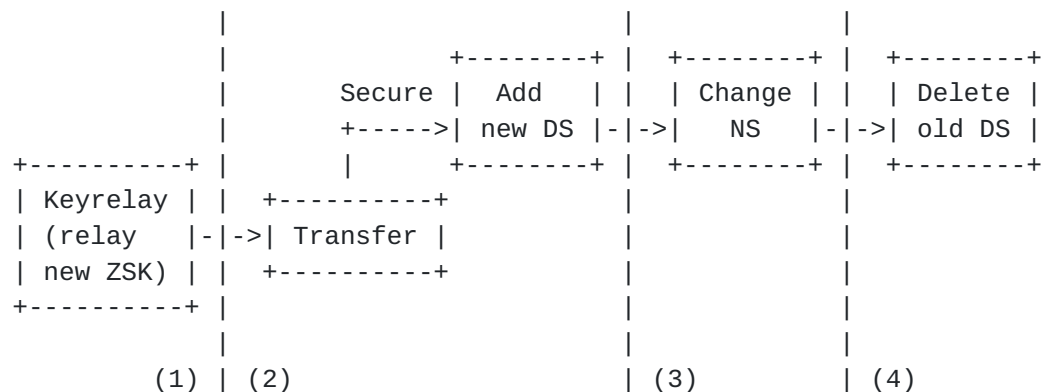origin at the losing or gaining operator's version of the zone.

The final third step can be started after all instances of DNSKEY
RRSets containing the old KSK have expired. This also requires that
no queries are directed to the name servers of the losing operator.
The latter can be assumed after the TTL of the (old) NS RRSet at the
parent has passed (no delegations to old infrastructure) and
subsequently the TTL of the NS RRSet at the losing operator's zone
has also passed (no refresh or overwrite of referral data by
authoritative data from the child zone). Since a DNSKEY RRSet might
have been sent as part of a DNS response just before this expiry,
this DNSKEY RRSet's (originating from the losing operator, containing
old KSK and new ZSK) TTL must pass, too. In total, counting from the
appearance of the new NS RRSet in the parent zone, the sum of the TTL
of the NS RRSet at the parent, the TTL of the NS RRSet at the child
(losing operator) and the TTL of the DNSKEY RR at the losing operator
determines the earliest point in time for proceeding with the next
step.

The third step will be removing the old KSK from the domain object as
part of the KSK rollover. The parent zone will then publish a single
DS RR pointing to the new KSK only. As soon as the new DS RRSet will
be the only one present in caches, the losing operator may cease to
serve the zone. The gaining operator, in turn, can remove the old
ZSK from its apex DNSKEY RRSet, not without re-signing this RRSet
afterwards.

3.2. Changing DNS operator and registrar

Should the DNS operator change involve a registrar change the
procedure described in the previous paragraph can be followed with
one minor change. The first step, adding the new KSK to be added as
DS in the parent zone , will be immediately preceded by a transfer
initiated by the gaining registrar. After the transfer step will
have been executed, the gaining registrar will be the sponsoring
entity for a domain object that continues to refer to the losing
operator's infrastructure and therefore probably also the losing

registrar's data untill the the other steps in the procedure have
completed.

```
              |                     |            |
              |           +--------+ |  +--------+ |  +--------+
              |    Secure |  Add   | |  | Change | |  | Delete |
              |    +----->| new DS |-|->|   NS   |-|->| old DS |
  +----------+ |    |     +--------+ |  +--------+ |  +--------+
  | Keyrelay | |  +----------+       |            |
  | (relay   |-|->| Transfer |       |            |
  | new ZSK) | |  +----------+       |            |
  +----------+ |                     |            |
              |                     |            |
        (1) | (2)                   | (3)        | (4)
```

   (1): Gaining operator configures zone with old ZSK included and
        signs
   (2): Loosing operator recieves new ZSK, includes ZSK in zone and
        resigns
   (3): Wait max TTL old DNSKEY RRset after seeing new ZSK in old
        zone and DS TTL
   (4): Wait TTL NS RRset old zone after NS change

   Figure: Changing DNS operator and registrar

3.3. Losing operator not participating

   The procedure described in the previous sections depends upon the
   losing DNS operator's participation to incorporate the new ZSK into
   their DNSKEY RRSet. Should the losing operator not participate,
   reasons for this being beyond the scope of this document, the gaining
   operator can notice this after waiting a reasonable amount of time
   after executing the first step.

The only known working way to avoid validation failures in this case
is to declare the zone insecure by removing the DS RR from the parent
zone. Some timing considerations are still due. After the parent
has stopped publishing the DS RRSet, and at least one DS TTL has
passed, the registered NS RRSet can be changed from the losing
operator's to the gaining operator's infrastructure. The gaining
operator's key material can be registered in a second step only after
the maximum of the TTL values of the parent's and the (losing
operator's) child's NS RRSet has passed, again counting from the
appearance of the NS RRSet in the parent zone. At this point, the
zone's security status is back to "secure". When an attempt has been
made to have the losing operator cooperate by including the old ZSK
in the new zone's DNSKEY RRset, this ZSK can be removed from the zone
one DS TTL after the old DS has been removed from the parent zone.

```
                   |                   |           |
   +----------+    |                   |           |
   | Keyrelay |    |   +----------+     |           |
   | (relay   |-|->| Transfer |     |           |
   | new ZSK) |    |   +----------+     |           |
   +----------+    |         |      +--------+ |  +--------+ |  +--------+
                   |   +------>| Delete |-|->| Change |-|->|  Add   |
                   |     Insecure | old DS | |  |   NS   | |  | new DS |
                   |              +--------+ |  +--------+ |  +--------+
                   |                   |           |
             (1) | (2)                 | (3)       | (4)
```
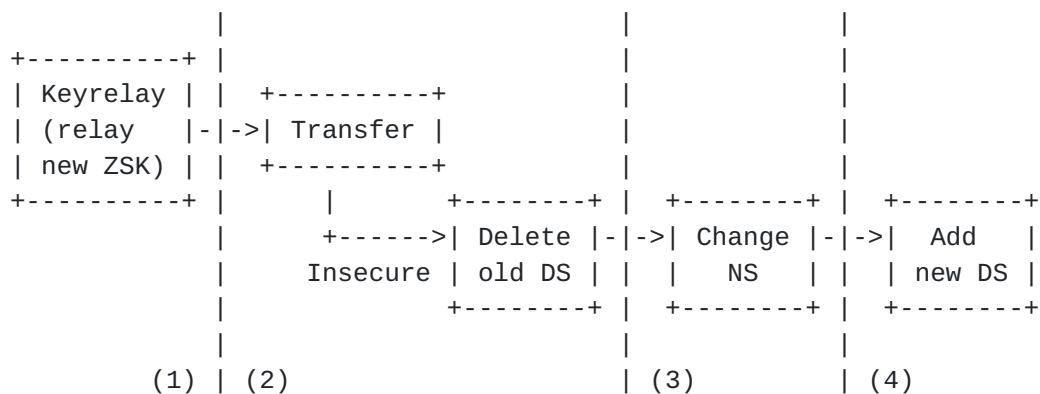
(1): Gaining operator configures zone with old ZSK included and
      signs
(2): Loosing operator is uncooperative
(3): Wait TTL DS RRset parent zone
(4): Wait TTL NS RRset old zone after NS change

 Figure: Loosing operator uncooperative, going insecure

[4]. Security Considerations

Since the procedure described in this document is incompatible with a
DNSKEY algorithm rollover during the operator change, it may
encourage the use of the same algorithm across all operators
involved. This could essentially limit the algorithm agility from an
operational perspective. Concerted action might be advised should
that preferred algorithm no longer be appropriate.

Preferring insecure state over validation failure is a judgement that
should be revisited, especially in the light of emerging application
protocols that will ignore unsigned or unvalidated DNS data.

As with a regular ZSK key rollover there is an odd chance that RRSets with larger TTL values than the DNSKEY and NS RRSets, which dominate the timing considerations, stay in a validator's cache. Any attempt to revalidate these would lead to validation failures due to the unavailability of the old ZSK.

5. IANA Considerations

This document does not request any IANA action.

6. Acknowledgements

The review, comments, and contributions by James Galvin and Paul Wouters are much appreciated. Special thanks go to the authors and contributors of [RFC4641] and [RFC6781] for detailed work on key rollovers.

7. References

7.1. Normative References

[RFC2606]   Eastlake, D.E. and A. Panitz, "Reserved Top Level DNS
            Names", BCP 32, RFC 2606, June 1999.

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

[RFC4641]   Kolkman, O. and R. Gieben, "DNSSEC Operational Practices",
            RFC 4641, September 2006.

[RFC5735]   Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses",
            RFC 5735, January 2010.

[RFC6781]   Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC
            Operational Practices, Version 2", RFC 6781, December
            2012.

7.2. 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.

[RFC2181]   Elz, R. and R. Bush, "Clarifications to the DNS
            Specification", RFC 2181, July 1997.

   [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.

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

Appendix A. Document Revision History

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

A.1. -00

   1. Initial document.

A.2. -01

   Expanded on the assumptions and requirements.

   Added initial version of the process description.

A.3. -02

   Split requirements and assumptions.

   Declared sticky resolvers a non-issue.

   Increased level of detail of discussion of TTL expiration between
   updates.

   Added early security considerations.

A.4. **-03**

   Removed redundant requirement.

   Reclassified 'sticky' resolver issue to assumption.

   Explictly assume secure path between registrar and registry.

   Clarified rollover in progress.

A.5. **-04**

   Clarified operational practice in the introduction.

   Added timing considerations for going through insecure.

   Elaborated on combined transfer and operator change.

   Expanded security considerations section.

   Elevated 4033, 4641, and 4641bis to normative references; relaxed
   1034, 1035, and 2181 to informative.

A.6. **-05**

   Added 6781 as normative reference

   Updated authors

A.7. **-06**

   Change reference in text from 4146bis to 6781

   Added ascii art

   Replaced communication of ZSK from entry in registration DB to
   generic communication channel

   Added text to 3.3 to remove old ZSK from new zone

Authors' Addresses

Peter Koch
DENIC eG
Kaiserstrasse 75-77
Frankfurt 60329
DE

Phone: +49 69 27235 0
Email: pk@DENIC.DE

Marcos Sanz
DENIC eG
Kaiserstrasse 75-77
Frankfurt 60329
DE

Phone: +49 69 27235 0
Email: sanz@DENIC.DE

Antoin Verschuren
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

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