Domain Name System (DNS) Security Key Rollover
------ ---- ------ ----- ------- --- --------
<**[draft-ietf-dnsop-rollover-00.txt](#)**>

Mark Andrews, Donald E. Eastlake 3rd

Status of This Document

   This draft, file name [draft-ietf-dnsop-rollover-00.txt](#), is intended
   to be become a Proposed Standard RFC. Distribution of this document
   is unlimited. Comments should be sent to the Domain Name Server
   Operations working group mailing list <dnsop@cafax.se> or to the
   authors.

   This document is an Internet-Draft and is in full conformance with
   all provisions of [Section 10 of RFC 2026](#). Internet-Drafts are
   working documents of the Internet Engineering Task Force (IETF), its
   areas, and its working groups. Note that other groups may also
   distribute working documents as Internet-Drafts.

   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."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

Abstract

   Deployment of Domain Name System (DNS) security with good cryptologic
   practice will involve large volumes of key rollover traffic. A
   standard format and protocol for such traffic will be necessary for
   this to be practical and is specified herein.

   [Note: The previous version of this draft was [draft-ietf-dnsind-](#)
   [rollover-00.txt](#) and before that [draft-ietf-dnssec-rollover-00.txt](#).]

Table of Contents

[1](#). **Introduction**

   The Domain Name System (DNS) [RFC 1034, 1035] is the global
   hierarchical replicated distributed database system for Internet
   addressing, mail proxy, and other information. The DNS has been
   extended to include digital signatures and cryptographic keys as
   described in [[RFC 2535](#)].

   The principle security service provided for DNS data is data origin
   authentication. The owner of each zone signs the data in that zone
   with a private key known only to the zone owner. Anyone that knows
   the corresponding public key can then authenticate that zone data is
   from the zone owner. To avoid having to preconfigure resolvers with
   all zone's public keys, keys are stored in the DNS with each zone's
   key signed by its parent (if the parent is secure).

   To obtain high levels of security, keys must be periodically changed,
   or "rolled over". The longer a private key is used, the more likely
   it is to be compromised due to cryptanalysis, accident, or treachery
   [[RFC 2541](#)].

   In a widely deployed DNS security system, the volume of update
   traffic will be large. Just consider the .com zone. If only a few
   percent of its children are secure and change their keys only once a
   year, you are talking about hundreds of thousands of new child public
   keys that must be securely sent to the .com manager to sign and
   return with their new parent signature. And when .com rolls over its
   private key, it will need to send hundred of thousands of new
   signatures on the existing child public keys to the child zones.

   It will be impractical to handle such update volumes manually on a
   case by case basis. The bulk of such key rollover updates must be
   automated.

   The key words "MUST", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY"
   in this document are to be interpreted as described in [[RFC 2119](#)].

[2](#). **Key Rollover Scenario**

   Although DNSSEC provides for the storage of other keys in the DNS for
   other purposes, DNSSEC zone keys are included solely for the purpose
   of being retrieved to authenticate DNSSEC signatures. Thus, when a
   zone key is being rolled over, the old public key should be left in
   the zone, along with the addition of the new public key, for as long
   as it will reasonably be needed to authenticate old signatures that
   have been cached or are held by applications. Similarly, old parent
   SIGs should be retained for a short time after a parent KEY(s) roll

over and new parent SIGs have been installed.

M. Andrews, D. Eastlake 3rd   [Page 3]

If DNSSEC were universally deployed and all DNS server's clocks were
synchronized and zone transfers were instantaneous etc., it might be
possible to avoid ever having duplicate old/new KEY/SIG RRsets due to
simultaneous expiration of SIGs everywhere in the DNS. But these
assumptions do not hold. Security aware DNS servers decrease the TTL
of secure RRs served as the expiration of their authenticating SIG(s)
approaches but some dithered fudge must generally be left due to
clock skew, rounding, RR retention by applications, and the like.
Retaining old KEYs for a while after rolling over to new KEYs will be
necessary in practical cases.

Assume a secure middle zone with a s parent and a secure child wishes
to role over its KEY RRset. This RRset would probably be one KEY RR
per crypto algorithm used to sign the zone, but for this scenario, we
will simply assume it is one KEY RR. The old KEY RR and two SIG RRs,
one signed by the parent and one signed by the zone itself, will
exist at the apex of the middle zone. (These RRs may also exist at
the leaf node for this zone in its parent if the parent chooses to
store them there.) The contents of this middle zone and the zone KEY
RRs of its secure child will have SIGs under the old key.

The middle zone owner needs to communicate with its parent to obtain
a new parental signature covering both the old and new KEY RRs and a
parental signature covering just the new KEY RR. The signature on
both is needed so the old KEY can be retain for the period it might
be needed to authenticate old SIGs. The middle zone would probably
want to obtain these in advance so that it can install them at the
right time along with its new SIG RRs covering the content of its
zone. Finally, it needs to give new SIG RRs to its child that cover
both its KEY RRs and the child's KEY RRs so it must signal its
children to ask for such SIG RRs.

The table below illustrates what happens during this rollover
scenario:

```
     BEFORE ROLLOVER          SHORTLY AFTER            AFTER ROLLOVER

   p.x     KEY       P1     p.x     KEY       P1     p.x     KEY       P1
   p.x     SIG(KEY) P1      p.x     SIG(KEY) P1      p.x     SIG(KEY) P1
   p.x     SIG(KEY) GP      p.x     SIG(KEY) GP      p.x     SIG(KEY) GP

   m.p.x   KEY       M1     m.p.x   KEY       M2     m.p.x   KEY       M2
   m.p.x   SIG(KEY) P1      m.p.x   KEY       M1     m.p.x   SIG(KEY) P1
   m.p.x   SIG(KEY) M1      m.p.x   SIG(KEY) P1      m.p.x   SIG(KEY) M2
                            m.p.x   SIG(KEY) M2

   c.m.p.x KEY       C1     c.m.p.x KEY       C1     c.m.p.x KEY       C1
   c.m.p.x SIG(KEY) M1      c.m.p.x SIG(KEY) M2      c.m.p.x SIG(KEY) M2
   c.m.p.x SIG(KEY) C1      c.m.p.x SIG(KEY) M1      c.m.p.x SIG(KEY) C1
                            c.m.p.x SIG(KEY) C1

       p = parent, m = middle, c = child, GP = grandparent key
       P* = parent key, M* = middle zone key, C* = child key
```

3. Rollover Operation

Rollover operations use a DNS request syntactically identical to the
UPDATE request [RFC 2136] except that the operation code is ROLLOVER,
which is equal to (TBD), and use a new variation of NOTIFY [RFC
1996]. Considerations are given below.

All rollover operations involve significant amounts of cryptographic
calculations. Appropriate rate limiting SHOULD be applied to avoid
denial of service attacks.

[This draft does not consider cross-certification key rollover.]

3.1 Rollover to Parent

A zone rolling over its KEY RRset sends an upward ROLLOVER request to
its parent. Actually, it will normally do two upward ROLLOVERs, one
for a combined KEY RRset of its old and new KEYs and one for just its
new KEY RRset, as discussed above.

The server selection algorithm in [RFC 2136] section 4 should be used
for the retrieval of SRV RRs [RFC 2782] using the service name (tbd)
to determine which host(s) to which the ROLLOVER request is sent. A
child needs to be configured with or determine the name of its
parent.

The ROLLOVER request Zone should be specified in the Zone section.

The request Update section has the new KEY RRset on which the parent
signature is requested along with the requesting zone's SIG(s) under
its old KEY(s) as RRs to be "added" to the parent zone.   The
inception and expiration times in this child SIG or SIGs are the
requested inception and expiration times for the new parent SIG(s).
The "prerequisites" section has the old child KEY RRset with the
parent SIG (see next paragraph).

An upward ROLLOVER request MUST be signed and if not signed a BADAUTH
response generated.   The signature MUST be a SIG(0) using the
previous zone KEY, so the parent can validate it, or be under a valid
TSIG key [RFC 2845] arranged with the parent.   Including the
"prerequisite" section as specified above enables a parent that keeps
no record of its children's KEYs to still authenticate a child's
ROLLOVER request based on the old child KEY because the parent is
presented with its own SIG on the old KEY.

If the ROLLOVER command is erroneous or violates parental policy, an
Error response is returned.   If a parent retains copies of its
children's KEYs, it may use that knowledge to validate ROLLOVER
request SIGs and ignore the "prerequisites" section.

If the ROLLOVER command is OK and the parent can sign online, its
response MAY include the new parent SIG(s) in the response Update
section.   This response MUST be sent to the originator of the
request.

If the parent can not sign online in a reasonable length of time, it
should return a response with an empty Update section and queue the
SIG(s) calculation request.   This response MUST be sent to the
originator of the request.

ROLLOVER response messages MUST always include, in the Additional
Information section, the actual parent's SOA signed with a key the
child should recognize (see section 4 below).

Regardless of whether the server has sent the new signatures above,
it MUST, once it has calculated the new SIG(s), send a ROLLOVER to
the child zone using the DNS port (53) and host determined as
follows: the server selection algorithm defined in RFC 2136, Section
4, for updates to the child zone is used to fetch SRV RRs for service
name (tbd).   This ROLLOVER reqeust contains the KEY RR set that
triggered it and the new SIG(s).   There are several reasons for
sending the ROLLOVER response regardless of whether the new SIG RR(s)
were sent in the original response.   One is to provide an indication
to the operators of the zone in the event someone is trying to hijack
the zone.   Another is that this maximizes the probability of the
response getting through.

Although the parent zone need not hold or serve the child's key, if

it does, the ROLLOVER command request MAY automatically update the
parent zone.

This document does not cover the question of parental policy on key
rollovers. Parents may have restrictions on how far into the future
they will sign KEY RRsets, what algorithms or key lengths they will
support, may require payment for the service, etc. The signing of a
future KEY by a parent is, to some extent, a granting of future
authoritative existence to the controller of the child private key
even if the child zone ownership should change. The only effective
way of invalidating such future signed child public keys would be for
the parent to roll over its key(s), which might be a very expensive
operation.

3.2 Rollover to Children

When a secure zone is going to rollover its key(s), it needs to re-
sign the zone keys of any secure children under its new key(s). The
parent simply NOTIFies the children via a rollover NOTIFY [RFC 1996]
that the parent KEY(s) have changed. The child then proceeds to do
an upward ROLLOVER request, as described in 3.1 above to obtain the
new parental SIG(s).

A rollover NOTIFY is a NOTIFY request [RFC 1996] that has a QTYPE of
SIG and the owner name of the child zone. The answer section has the
current parent SOA signed by a key the child will know (see section
4).

A rollover NOTIFY MUST be signed and if not signed a BADAUTH response
generated. The signature MUST be under the previous parental zone
KEY, so the child can validate it, or under a valid TSIG key [RFC
2845] negotiated between parent and child.

The rollover NOTIFY MUST be send by using the the nameserver
selection algorithm defined in RFC 2136, Section 4, to fetch SRV RRs
for the (tbd) named service. Servers for the child zone receiving a
rollover NOTIFY query will forward the rollover NOTIFY in the same
manner as an UPDATE is forwarded except that they will forward using
SRV RRs as above.

Unless the rollover server for the zone master is configured to
initiate an automatic ROLLOVER it MUST seek to inform its operators
that a rollover NOTIFY request has been received. This could be done
by a number of methods including generating a log message, generating
an email request to the child zone's SOA RNAME or any other method
defined in the server's configuration for the zone. The default
SHOULD be to send mail to the zone's SOA RNAME. As with all rollover

operations, care should be taken to rate limit these messages so

prevent them being used to facilitate a denial of service attack.

Once the message has been sent (or suppressed if so configured) to
the child zone's administrator the master server for the child zone
is free to respond to the rollover NOTIFY request.

[4](#). Secure Zone Cuts and Joinders

There are two other events that have some similarity to key rollover.

The first is when a secure zone the is more than one level deep has a
zone cut introduced inside it. For example, assume zone example.com
has a.b.c.example.com, d.b.c.example.com and e.example.com in it. A
zone cut could be introduced such that b.c.example.com became a
separate child zone of example.com. A real world example would be a
company that structures its DNS as host.branch.company.example. It
might start out will all of these names in one zone but later decide
to delegate all or some of the branches to branch zone file
maintainers.

The second is when a secure zone absorbs a child zone eliminating a
zone cut. This is simply the inverse of the previous paragraph.

From the point of view of the parent zone above the splitting zone or
above the upper of the two combining zones, there is no change. When
a zone is split by introducing a cut, the newly created child must be
properly configured.

However, from the point of view of a child of the splitting zone
which becomes a grandchild or a grandchild that becomes a child due
to joinder, there is a change in parent name. Therefore, in the
normal case, there is a change in parent KEY(s). Unless the entity
that handles rollovers for the zone whose parent name has changed is
appropriately updated, future automated rollovers will fail because
they will be sent to the old parent.

For this reason and so that other consistency checks can be made, the
parent SOA and SIG(SOA) are always included in the Answer section of
rollover NOTIFY requests and in ROLLOVER responsess. For automated
rollover to the new cut or joined state to work, these SOAs must be
signed with old KEY(s) of the former parent so the signatures can be
validated by the zone whose parent name is changing. In the case of
a joinder, if the private key of the pinched out middle zone is not
available, then manual update of the former grandchild, now child,
will be necessary. In the case of introducing a cut, operational
coordination with the former parent, now grandparent, signing the
initial updates to the former child, now grandchild, will be needed

to automate the reconfiguration of the zones.

[5](#). **Security Considerations**

   The security of ROLLOVER or UPDATE requests is essential, otherwise
   false children could steal parental authorization or a false parent
   could cause a child to install an invalid signature on its zone key,
   etc.

   A ROLLOVER request can be authenticated by request SIG(s)under the
   old zone KEY(s) of the requestor [[RFC 2535](#)]. The response SHOULD
   have transaction SIG(s) under the old zone KEY(s) of the responder.

   Alternatively, if there is a prior arrangement between a child and a
   parent, ROLLOVER requests and responses can be secured and
   authenticated using TSIG [[RFC 2845](#)].

   A server that implements online signing SHOULD have the ability to
   black list a zone and force manual processing or demand that a
   particular signature be used to generate the ROLLOVER request. This
   is to allow ROLLOVER to be used even after a private key has been
   compromised.

[6](#). **IANA Considerations**

   The DNS operation code (TBD) is assigned to ROLLOVER.

   The Service Name (TBD) is assigned to the DNS key rollover service.

   There are no other IANA considerations in this document.

References

    [RFC 1034] - "Domain names - concepts and facilities", P.
    Mockapetris, 11/01/1987.

    [RFC 1035] - "Domain names - implementation and specification", P.
    Mockapetris, 11/01/1987.

    [RFC 1996] - "A Mechanism for Prompt Notification of Zone Changes
    (DNS NOTIFY)", P. Vixie, August 1996.

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

    [RFC 2136] - "Dynamic Updates in the Domain Name System (DNS
    UPDATE)", P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound. April
    1997.

    [RFC 2535] - "Domain Name System Security Extensions", D. Eastlake.
    March 1999.

    [RFC 2541] - "DNS Security Operational Considerations", D. Eastlake.
    March 1999.

    [RFC 2782] - "A DNS RR for specifying the location of services (DNS
    SRV)", A. Gulbrandsen, P. Vixie, L. Esibov. February 2000.

    [RFC 2845] - "Secret Key Transaction Authentication for DNS (TSIG)",
    P. Vixie, O. Gundmundsson, D. Eastlake, B. Wellington. May 2000.

Authors Address

    Donald E. Eastlake 3rd
    Motorola
    140 Forest Avenue
    Hudson, MA 01749 USA

    Telephone: +1 978-562-2827 (h)
                  +1 508-261-5434 (w)
    FAX: +1 508-261-4447 (w)
    EMail: Donald.Eastlake@motorola.com

    Mark Andrews
    Internet Software Consortium
    1 Seymour Street
    Dundas Valley, NSW 2117
    AUSTRALIA

    Telephone: +61-2-9871-4742
    Email: Mark_Andrews@iengines.com

Expiration and File Name

    This draft expires in December 2000.

    Its file name is draft-ietf-dnsop-rollover-00.txt.