Network Working Group
Internet-Draft
Expires: January 7, 2005

J. Ihren
Autonimica AB
O. Kolkman
RIPE NCC
B. Manning
EP.net
July 9, 2004

An In-Band Rollover Algorithm and a Out-Of-Band Priming Method for DNS Trust Anchors.
draft-kolkman-dnsext-dnssec-in-band-rollover-00

Status of this Memo

Copyright Notice

Abstract

The DNS Security Extensions (DNSSEC) works by validating so called chains of authority. The start of these chains of authority are usually public keys that are anchored in the DNS clients, the so called trust anchors.

This memo describes a method how these client trust anchors can be replaced using the DNS validation and querying mechanisms (in-band) if the key pairs used for signing by zone owner are rolled.

This memo also describes a method to establish the validity of trust anchors for initial configuration, or priming, using out of band mechanisms.

Table of Contents

1. Terminology

The key words "MUST", "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in RFC2119 [1].

The term "zone" refers to the unit of administrative control in the Domain Name System. In this document "name server" denotes a DNS name server that is authoritative (i.e. knows all there is to know) for a DNS zone. A "zone owner" is the entity responsible for signing and publishing a zone on a name server. The terms "authentication chain", "bogus", "trust anchors" and "Island of Security" are defined in [4]. Throughout this document we use the term "resolver" to mean "Validating Stub Resolvers" as defined in [4].

We use the term "security apex" as the zone for which a trust anchor has been configured and which is therefore, by definition, at the root of an island of security. The configuration of trust anchors is a client side issue therefore a zone owner may not always know if their zone has become a security apex.

A "stale anchor" is a trust anchor (a public key) that relates to a key that is not used for signing. Since trust anchors indicate that a zone is supposed to be secure a validator will mark the all data in an island of security as bogus when all trust anchors become stale.

It is assumed that the reader is familiar with public key cryptography concepts [REF: Schneier Applied Cryptography] and is able to distinguish between the private and public parts of a key based on the context in which we use the term "key", if there is a possible ambiguity we will explicitly mention if a private or a public part of a key is used.

The term "administrator" is used loosely throughout the text. In some cases an administrator is meant to be a person, in other cases the administrator may be a process that has been delegated certain responsibilities.

1.1 Key Signing Keys, Zone Signing Keys and Secure Entry Points.

Although the DNSSEC protocol does not make a distinction between different keys the operational practice is that a distinction is made between zone signing keys and key signing keys. A key signing key is used to exclusively sign the DNSKEY Resource Record (RR) set at the apex of a zone and the zone signing keys sign all the data in the zone (including the DNSKEY RR set at the apex).

Keys that are intended to be used as the start of the authentication

chain for a particular zone, either because they are pointed to by a parental DS RR or because they are configured as a trust anchor, are called Secure Entry Point (SEP) keys. In practice these SEP keys will be key signing keys.

In order for the mechanism described herein to work the keys that are intended to be used as secure entry points MUST have the SEP [2] flag set. In the examples it is assumed that keys with the SEP flag set are used as key signing keys and thus exclusively sign the DNSKEY RR set published at the apex of the zone.

2. Introduction and Background

When DNSSEC signatures are validated the resolver constructs a chain of authority from a pre-configured trust anchor to the DNSKEY Resource Record (RR), which contains the public key that validates the signature stored in a RRSIG RR. DNSSEC is designed so that the administrator of a resolver can create multiple islands of security by configuring multiple trust anchors.

It is expected that resolvers will have more than one trust-anchor configured. Although there is no deployment experience it is not unreasonable to expect resolvers to be configured with a number of trust anchors that varies between order 1 and order 1000. Because zone owners are expected to roll their keys, trust-anchors will have to be maintained in order not to become stale.

Since there is no global key maintenance policy for zone owners and there are no mechanisms in the DNS to signal the key maintenance policy it may be very hard for resolvers administrators to keep their set of trust anchors up to date. For instance, if there is only one trust anchor configured and the key maintenance policy is clearly published, through some out of band trusted channel, than a resolver administrator can probably keep track of key rollovers and update the trust anchor annually. With more than 100 different policies all published through different channels this soon becomes an unmanageable problem.

In Section 3 this memo sets out a lightweight, in-DNS, mechanism to track key rollovers and modify the trust-anchor's accordingly. The algorithm is stateless and does not need protocol extensions.

In Section 4 we describe a method [Editors note: for now only the frame work and a set of requirements] to install trust anchors. This method can be used at first configuration or when the trust anchors became out of sync with the keys published by a zone owner.

The choice for which domains trust anchors are to be configured is a

local policy issue. So is the choice which trust anchors has prevalence if there are multiple chains of trust to a given piece of DNS data (e.g. when a domain and its sub-domain both have a trust anchor configured). Both issues are out of the scope of this document.

3. M-N Trust Anchor Rollover.

3.1 The Rollover

When a key pair is replaced all signatures (in DNSSEC these are the RRSIG records) created with the old key will be replaced by new signatures created by the new key. Access to the new public key is needed to verify these signatures.

Since a zone signing keys are in "the middle" of a chain of authority the can be verified using the signature made by a key signing key. Rollover is therefore transparent to validators. But if a key signing key is rolled a resolver can determine its authenticity by either following the authorization chain from the parents DS RR, an out-of-DNS authentication or by relying on other trust anchors known for the zone in which the key is rolled. The M-N trust anchor rollover mechanism, described below, is based on using existing trust anchors to verify a subset of the available signatures.

Our example pseudo zone below contains a number of key signing keys numbered 1 through Y and two zone signing keys A and B. During a key rollover key 2 is replaced by key Y+1. The zone content changes from:

```
example.com.  DNSKEY key1
example.com.  DNSKEY key2
example.com.  DNSKEY key3
...
example.com.  DNSKEY keyN

example.com.  DNSKEY keyA
example.com.  DNSKEY keyB

example.com.  RRSIG DNSKEY ... (key1)
example.com.  RRSIG DNSKEY ... (key2)
example.com.  RRSIG DNSKEY ... (key3)
...
example.com.  RRSIG DNSKEY ... (keyY)
example.com.  RRSIG DNSKEY ... (keyA)
example.com.  RRSIG DNSKEY ... (keyB)
```

to:

```
example.com.  DNSKEY key1
example.com.  DNSKEY key3
...
example.com.  DNSKEY keyY
example.com.  DNSKEY keyY+1

example.com.  RRSIG DNSKEY ... (key1)
example.com.  RRSIG DNSKEY ... (key3)
...
example.com.  RRSIG DNSKEY ... (keyY)
example.com.  RRSIG DNSKEY ... (keyY+1)
```

When the rollover becomes visible to the verifying stub resolver it will be able to verify the RRSIGs associated with key1, key3 ... keyY. There will be no RRSIG by key2 and the RRSIG by keyY+1 will not be used for validation, since that key is previously unknown and thereby not trusted.

Note that this example is simplified. Because of operational considerations described in [5] having a period where the two key signing keys are available is actually a good idea.

3.2 The Algorithm.

The M-N trust anchor rollover algorithm applies as follows. If for a particular zone

- at least M public keys from the trust anchors directly verify the related RRSIGs over the DNSKEY RRset. (the M criterion)

and if

- the number of element in the set difference between 'the set of keys from the DNSKEY RRset with the SEP bit set' and 'the set of keys configured as trust anchors' is smaller or equal than N. (the N criterion)

then all the trust-anchors for the particular zone replaced by the keys from the zones DNSKEY RR set that have the SEP flag set

The choices for the rollover acceptance policy parameters M and N are left to the administrator of the resolver. To be certain that a rollover is picked up by resolvers running this algorithm zone owners should only roll 1 SEP key at a time. That way they comply to the most strict rollover acceptance policy of N=1. The value of M is limited by the amount of SEP keys a zone owner publishes. If the policy of the zone owner is to use Y SEP keys than the value of M should be M<=Y-N.

If the rollover acceptance policy is M=1 then the result for the rollover in our example above should be that the local database of trusted keys is updated by removing key "key2" and adding key

"keyN+1" to the key store.

3.3 Implementation notes

The DNSSEC protocol ordains that all DNSKEYs should be self-signed. The implementation should check this. In order to be resilient against failures the implementation should collect the DNSKEY RRsets from (other) authoritative servers if verification of the self signatures fails.

The algorithm SHOULD only be applied to algorithms, as represented in the algorithm field in the DNSKEY/RRSIG [3], that the resolver is aware of. In other words the SEP keys of unknown algorithms should not be used when calculating the set difference for the N parameter and the SEP keys of unknown algorithm should not be entered as trust anchors.

If a the M criterion is not met then the set of trust-anchors is out of sync with the SEP keys in the DNSKEY RRset and some or all of the trust-anchors are stale. This condition should be flagged. The most appropriate action is human audit possibly followed by re-priming (Section 4) the keyset.

An implementation should regularly probe the the authoritative nameservers for new keys. Since there is no mechanism to publish rollover frequencies this document RECOMMENDS zone owners not to roll their key signing keys more often than once per month and resolver administrators to probe for key rolls (and apply the M-N algorithm) not less often than once per month. If the rollover frequency is higher than the probing frequency than trust anchors may become stale. The exact relation between the frequencies depends on the amount of SEP keys rolled by the zone owner and the value M configured by the resolver administrator.

In all the cases below a transaction where M-N algorithm does not validate should be considered bad (i.e. possibly spoofed or otherwise corrupted data). The most appropriate action is human audit.

3.4 Possible transactions.

3.4.1 Single DNSKEY replaced.

This is probably the most typical transaction on the zone owners part. The result should be that if the M-N algorithm validates then the key store is updated by removal of the old key and addition of the new key. Note that if the DNSKEY RRset contains exactly M keys replacement of keys is not possible.

3.4.2 Addition of a new DNSKEY (no removal)

If M-N algorithm validates then the new key is added to the key store. Not more than N keys can be added at once.

3.4.3 Removal of old DNSKEY (no addition).

If the M-N algorithm validates then the old key is removed from the key store. Note that it is not possible to reduce the keyset to a size smaller than M.

3.4.4 Multiple DNSKEYs replaced.

Not more than N keys can be replace at one time. Since M keys need to validate the total number of SEP keys in the DNSKEY RRset is M+N.

Since the value of N is set by the resolvers local policy zone owners should assume N==1 in order to prevent a subset of the resolvers to become stale because they did not pick up the change.

3.4.5 Only some RRSIGs validate over an unchanged DNSKEY set.

This is a case where the M criterion may not be met, see Section 3.3.

3.5 No need for resolver-side overlap of old and new keys.

It is worth pointing out that there is no need for the resolver to keep state about old keys versus new keys. From the resolver point of view there are only trusted and not trusted keys. The reason is that the zone owner needs to do proper maintenance of RRSIGs regardless of the resolver rollover mechanism and hence must ensure that a key is not rolled out out the DNSKEY set until there cannot be any RRSIGs created by this key still legally cached.

Hence the rollover mechanism is stateless: as soon as the resolver (or in this case the rollover tracking utility) detects a change in the DNSKEY set with a sufficient number of matching RRSIGs the trusted key definition is immediately updated.

4. Bootstrapping automatic rollovers.

It is expected that with the ability to automatically roll trust anchors at trust points will follow a diminished unwillingness to roll these keys, since the risks associated with stale keys are minimized.

The problem of "priming" the trust anchors, or bringing them into sync (which could happen if a resolver is off line for a long period

in which a set of SEP keys in a zone 'evolve' away from its trust anchor configuration) remains.

For (re)priming we can rely on out of band technology and we propose the following framework.

4.1 Priming Keys.

If all the trust anchors roll somewhat frequently (on the order of months or at most about a year) then it will not be possible to design a device, or a software distribution that includes trust anchors, that after being manufactured is put on a shelf for several key rollover periods before being brought into use (since no trust anchors that were known at the time of manufacture remain active).

To alleviate this we propose the concept of "priming keys". Priming keys are ordinary DNSSEC Key Signing Keys with the characteristic that

- The private part of a priming key signs the DNSKEY RRset at the security apex, i.e. at least one RRSIG DNSKEY is created by a priming key rather than by an "ordinary" trust anchor
- the public parts of priming keys are not included in the DNSKEY RRset. Instead the public parts of priming keys are only available out-of-band.
- The public parts of the priming keys have a validity period. Within this period they can be used to obtain trust anchors.
- The priming key pairs are long lived (relative to the key rollover period.)

4.1.1 Bootstrapping trust-anchors using a priming key.

To install the trust-anchor for a particular security apex an administrator of a validating resolver will need to:

- query for the DNSKEY RR set of the zone at the security apex;
- verify the self signatures of all DNSKEYs in the RRset;
- verify the signature of the RRSIG made with a priming key -- verification using one of the public priming keys that is valid at that moment is sufficient;
- create the trust anchors by extracting the DNSKEY RRs with the SEP flag set.

The SEP keys with algorithms unknown to the validating resolver SHOULD be ignored during the creation of the trust anchors.

4.1.2 Distribution of priming keys.

The public parts of the priming keys SHOULD be distributed exclusively through out-of-DNS mechanisms. The requirements for a distribution mechanism are:

- it can carry the "validity" period for the priming keys;
- it can carry the self-signature of the priming keys;
- and it allows for verification using trust relations outside the DNS.

A distribution mechanism would benefit from:

- the availability of revocation lists;
- the ability of carrying zone owners policy information such as recommended values for "M" and "N" and a rollover frequency;
- and the technology on which is based is readily available.

[Editors Note:

X.509 technology is a logical candidate for the distribution of priming keys. The exact details need further research.

What we probably need a convention on the use id-ce-keyUsage, the id-ce-extKeyUsage (assignment of a KeyPurposeId) and other relevant field [6]. Is there a possibility to store the keyroll policy information? PKIX specialist are invited to give their input.

End of Editors note]

5. M-N algorithm vs Priming

There is overlap in the M-N algorithm and the Priming method. One could exclusively use the Priming method for maintaining the trust anchors. However the priming method probably relies on "non-DNS' technology and may therefore not be available for all devices that have a resolver.

6. Security Considerations.

6.1 M-N Algorithm Security Considerations

A clear issue for resolvers will be how to ensure that they track all rollover events for the zones they have configure trust anchors for. Because of temporary outages validating resolvers may have missed a rollover of a KSK. The parameters that determine the robustness against failures are: a long period between rollovers during which the KSK set is stable and validating resolvers can actually notice the change; the number of KSKs and the value of M.

With a large number of KSKs and a small value of M this operation becomes more robust since losing one key, for whatever reason, will not be crucial. Unfortunately the choice for the number of KSKs is a local policy issue for the zone owner while the choice for the parameter M is a local policy issue resolvers administrator.

Higher values of M increase the resilience against attacks somewhat; more signatures need to verify before a rollover is approved.

The M-N algorithm does not provide a revocation mechanism. In the case that the private keys of a zone owner are compromised the culprit may use these private keys to roll the trust anchors of (a subset of) the resolvers.

6.2 Priming Key Security Considerations

Since priming keys are not included in the DNSKEY RR set they are less sensitive to packet size constraints and can be chosen relatively large. The private parts are only needed to sign the DNSKEY RR set during the validity period of the particular priming key pair. Note that the private part of the priming key is used each time when a DNSKEY RRset has to be resigned in practice there shall little difference between the usage pattern of the private part of key signing keys and priming keys.

7. IANA Considerations.

NONE.

8. References

8.1 Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Kolkman, O., Schlyter, J. and E. Lewis, "Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag", RFC 3757, May 2004.

[3] Arends, R., "Resource Records for DNS Security Extensions", draft-ietf-dnsext-dnssec-records-04 (work in progress), September 2003.

8.2 Informative References

[4] Arends, R., Austein, R., Massey, D., Larson, M. and S. Rose, "DNS Security Introduction and Requirements", draft-ietf-dnsext-dnssec-intro-10 (work in progress), May 2004.

[5] Kolkman, O., "DNSSEC Operational Practices", draft-ietf-dnsop-dnssec-operational-practices-01 (work in progress), May 2004.

[6] Housley, R., Ford, W., Polk, T. and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999.

Authors' Addresses

Johan Ihren
Autonimica AB
Bellmansgaten 30
Stockholm SE-118 47
Sweden

EMail: johani@autonomica.se

Olaf M. Kolkman
RIPE NCC
Singel 256
Amsterdam 1016 AB
NL

Phone: +31 20 535 4444
EMail: olaf@ripe.net
URI: http://www.ripe.net/

Bill Manning
EP.net
Marina del Rey, CA 90295
USA

Appendix A. Acknowledgments

The present design for in-band automatic rollovers of DNSSEC trust anchors is the result of many conversations and it is no longer possible to remember exactly who contributed what.

In addition we've also had appreciated help from (in no particular order) Paul Vixie, Sam Weiler, Suzanne Woolf, Steve Crocker, Matt Larson and Mark Kosters.

Appendix B. Document History

This appendix will be removed if and when the document is submitted to the RFC editor.

The version you are reading is tagged as $Revision: 1.5 $.

Text between square brackets, other than references, are editorial comments and will be removed.

B.1 version 00

Kolkman documented the ideas provided by Ihren and Manning. In the process of documenting (and prototyping) Kolkman changed some of the details of the M-N algorithms working. Ihren did not have a change to review the draft before Kolkman posted;

Kolkman takes responsibilities for omissions, fuzzy definitions and mistakes.

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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