Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 9, 2020

S. Huque
P. Aras
Salesforce
J. Dickinson
Sinodun
J. Vcelak
NS1
D. Blacka
Verisign

March 8, 2020

Multi Signer DNSSEC models
draft-ietf-dnsop-multi-provider-dnssec-04

Abstract

Many enterprises today employ the service of multiple DNS providers to distribute their authoritative DNS service. Deploying DNSSEC in such an environment may present some challenges depending on the configuration and feature set in use. In particular, when each DNS provider independently signs zone data with their own keys, additional key management mechanisms are necessitated. This document presents deployment models that accommodate this scenario and describe these key management requirements. These models do not require any changes to the behavior of validating resolvers, nor do they impose the new key management requirements on authoritative servers not involved in multi signer configurations.

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 https://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 September 9, 2020.

Table of Contents

1. Introduction and Motivation 2
2. Deployment Models 3
2.1. Multiple Signer models 3
2.1.1. Model 1: Common KSK, Unique ZSK per provider 4
2.1.2. Model 2: Unique KSK and ZSK per provider 4
3. Validating Resolver Behavior 5
4. Signing Algorithm Considerations 6
5. Authenticated Denial Considerations 6
5.1. Single Method 7
5.2. Mixing Methods 7
6. Key Rollover Considerations 7
6.1. Model 1: Common KSK, Unique ZSK per provider 8
6.2. Model 2: Unique KSK and ZSK per provider 8
7. Using Combined Signing Keys 9
8. Use of CDS and CDNSKEY 9
9. Key Management Mechanism Requirements 10
10. DNS Response Size Considerations 10
11. IANA Considerations 11
12. Security Considerations 11
13. Acknowledgments 11
14. References 11
14.1. Normative References 11
14.2. Informative References 12
Authors' Addresses 13

1. Introduction and Motivation

RFC EDITOR: PLEASE REMOVE THE FOLLOWING PARAGRAPH BEFORE PUBLISHING:
The source for this draft is maintained in GitHub at:
https://github.com/shuque/multi-provider-dnssec

Many enterprises today employ the service of multiple DNS providers to distribute their authoritative DNS service. This allows the DNS service to survive a complete failure of any single provider. Additionally, enterprises or providers occasionally have requirements that preclude standard zone transfer techniques [RFC1995] [RFC5936] : either non-standardized DNS features are in use that are incompatible with zone transfer, or operationally a provider must be able to (re)sign DNS records using their own keys. This document outlines some possible models of DNSSEC [RFC4033] [RFC4034] [RFC4035] deployment in such an environment.

2. Deployment Models

If a zone owner is able to use standard zone transfer techniques, then the presence of multiple providers does not present any need to substantially modify normal deployment models. In these deployments there is a single signing entity (which may be the zone owner, one of the providers, or a separate entity), while the providers act as secondary authoritative servers for the zone.

Occasionally, however, standard zone transfer techniques cannot be used. This could be due to the use of non-standard DNS features, or due to operational requirements of a given provider (e.g., a provider that only supports "online signing".) In these scenarios, the multiple providers each act like primary servers, independently signing data received from the zone owner and serving it to DNS queriers. This configuration presents some novel challenges and requirements.

2.1. Multiple Signer models

In this category of models, multiple providers each independently sign and serve the same zone. The zone owner typically uses provider-specific APIs to update zone content at each of the providers, and relies on the provider to perform signing of the data. A key requirement here is to manage the contents of the DNSKEY and DS

RRset in such a way that validating resolvers always have a viable path to authenticate the DNSSEC signature chain no matter which provider is queried. This requirement is achieved by having each provider import the public Zone Signing Keys (ZSKs) of all other providers into their DNSKEY RRsets.

These models can support DNSSEC even for the non-standard features mentioned previously, if the DNS providers have the capability of signing the response data generated by those features. Since these responses are often generated dynamically at query time, one method is for the provider to perform online signing (also known as on-the-fly signing). However, another possible approach is to pre-compute

all the possible response sets and associated signatures and then algorithmically determine at query time which response set needs to be returned.

In the models presented, the function of coordinating the DNSKEY or DS RRset does not involve the providers communicating directly with each other. Feedback from several commercial managed DNS providers indicates that they may be unlikely to directly communicate since they typically have a contractual relationship only with the zone owner. However, if the parties involved are agreeable, it may be possible to devise a protocol mechanism by which the providers directly communicate to share keys.

The following descriptions consider the case of two DNS providers, but the model is generalizable to any number.

2.1.1. Model 1: Common KSK, Unique ZSK per provider

- Zone owner holds the KSK, manages the DS record, and is responsible for signing the DNSKEY RRset and distributing the signed DNSKEY RRset to the providers.

- Each provider has their own ZSK which is used to sign data.

- Providers have an API that owner uses to query the ZSK public key, and insert a combined DNSKEY RRset that includes both ZSKs and the KSK, signed by the KSK.

- Note that even if the contents of the DNSKEY RRset don't change,

the Zone owner of course needs to periodically re-sign it as signature expiration approaches. The provider API is also used to thus periodically redistribute the refreshed DNSKEY RRset.

- Key rollovers need coordinated participation of the zone owner to update the DNSKEY RRset (for KSK or ZSK), and the DS RRset (for KSK).

2.1.2. Model 2: Unique KSK and ZSK per provider

- Each provider has their own KSK and ZSK.
- Each provider offers an API that the Zone Owner uses to import the ZSK of the other provider into their DNSKEY RRset.
- DNSKEY RRset is signed independently by each provider using their own KSK.
- Zone Owner manages the DS RRset that includes both KSKs.

- Key rollovers need coordinated participation of the zone owner to update the DS RRset (for KSK), and the DNSKEY RRset (for ZSK).

3. Validating Resolver Behavior

The central requirement for both of the Multiple Signer models (Section 2.1) is to ensure that the ZSKs from all providers are present in each provider's apex DNSKEY RRset, and is vouched for by either the single KSK (in model 1) or each provider's KSK (in model 2.) If this is not done, the following situation can arise (assuming two providers A and B):

- The validating resolver follows a referral (delegation) to the zone in question.
- It retrieves the zone's DNSKEY RRset from one of provider A's nameservers.
- At some point in time, the resolver attempts to resolve a name in the zone, while the DNSKEY RRset received from provider A is still viable in its cache.

- It queries one of provider B's nameservers to resolve the name, and obtains a response that is signed by provider B's ZSK, which it cannot authenticate because this ZSK is not present in its cached DNSKEY RRset for the zone that it received from provider A.

- The resolver will not accept this response. It may still be able to ultimately authenticate the name by querying other nameservers for the zone until it elicits a response from one of provider A's nameservers. But it has incurred the penalty of additional roundtrips with other nameservers, with the corresponding latency and processing costs. The exact number of additional roundtrips depends on details of the resolver's nameserver selection algorithm and the number of nameservers configured at provider B.

- It may also be the case that a resolver is unable to provide an authenticated response because it gave up after a certain number of retries or a certain amount of delay. Or that downstream clients of the resolver that originated the query timed out waiting for a response.

Hence, it is important that the DNSKEY RRset at each provider is maintained with the active ZSKs of all participating providers. This ensures that resolvers can validate a response no matter which provider's nameservers it came from.

Details of how the DNSKEY RRset itself is validated differs. In model 1 (Section 2.1.1), one unique KSK managed by the Zone Owner signs an identical DNSKEY RRset deployed at each provider, and the signed DS record in the parent zone refers to this KSK. In model 2 (Section 2.1.2), each provider has a distinct KSK and signs the DNSKEY RRset with it. The Zone Owner deploys a DS RRset at the parent zone that contains multiple DS records, each referring to a distinct provider's KSK. Hence it does not matter which provider's nameservers the resolver obtains the DNSKEY RRset from, the signed DS record in each model can authenticate the associated KSK.

4. Signing Algorithm Considerations

DNS providers participating in multi-signer models need to use a common DNSSEC signing algorithm. This is because the current

specifications require that if there are multiple algorithms in the DNSKEY RRset, then RRsets in the zone need to be signed with at least one DNSKEY of each algorithm, as described in RFC 4035 [RFC4035], Section 2.2. If providers employ distinct signing algorithms, then this requirement cannot be satisfied.

5. Authenticated Denial Considerations

Authentiated denial of existence enables a resolver to validate that a record does not exist. For this purpose, an authoritative server presents, in a response to the resolver, NSEC (Section 3.1.3 of [RFC4035]) or NSEC3 (Section 7.2 of [RFC5155]) records. The NSEC3 method enhances NSEC by providing opt-out for signing insecure delegations and also adds limited protection against zone enumeration attacks.

An authoritative server response carrying records for authenticated denial is always self-contained and the receiving resolver doesn't need to send additional queries to complete the denial proof data. For this reason, no rollover is needed when switching between NSEC and NSEC3 for a signed zone.

Since authenticated denial responses are self-contained, NSEC and NSEC3 can be used by different providers to serve the same zone. Doing so however defeats the protection against zone enumeration provided by NSEC3. A better configuration involves multiple providers using different authenticated denial of existence mechanisms that all provide zone enumeration defense, such as pre-computed NSEC3, NSEC3 White Lies [RFC7129], NSEC Black Lies [BLACKLIES], etc. Note however that having multiple providers offering different authenticated denial mechanisms may impact how effectively resolvers are able to make use of the caching of negative responses.

5.1. Single Method

Usually, the NSEC and NSEC3 methods are used exclusively (i.e. the methods are not used at the same time by different servers). This configuration is prefered because the behavior is well-defined and it's closest to the current operational practice.

5.2. Mixing Methods

Compliant resolvers should be able to validate zone data when different authoritative servers for the same zone respond with different authentiated denial methods because this is normally observed when NSEC and NSEC3 are being switched or when NSEC3PARAM is updated.

Resolver software may be however designed to handle a single transition between two authenticated denial configurations more optimally than permanent setup with mixed authenticated denial methods. This could make caching on the resolver side less efficient and the authoritative servers may observe higher number of queries. This aspect should be considered especially in context of Aggresive Use of DNSSEC-Validated Cache [RFC8198].

In case all providers cannot be configured for a matching authentiated denial, it is advised to find lowest number of possible configurations possible across all used providers.

Note that NSEC3 configuration on all providers with different NSEC3PARAM values is considered a mixed setup.

6. Key Rollover Considerations

The Multiple Signer (Section 2.1) models introduce some new requirements for DNSSEC key rollovers. Since this process necessarily involves coordinated actions on the part of providers and the Zone Owner, one reasonable strategy is for the Zone Owner to initiate key rollover operations. But other operationally plausible models may also suit, such as a DNS provider initiating a key rollover and signaling their intent to the Zone Owner in some manner.

The descriptions in this section assume that KSK rollovers employ the commonly used Double Signature KSK Rollover Method, and that ZSK rollovers employ the Pre-Publish ZSK Rollover Method, as described in detail in [RFC6781]. With minor modifications, they can also be easily adapted to other models, such as Double DS KSK Rollover or Double Signature ZSK rollover, if desired.

6.1. Model 1: Common KSK, Unique ZSK per provider

- Key Signing Key Rollover: In this model, the two managed DNS providers share a common KSK which is held by the Zone Owner. To initiate the rollover, the Zone Owner generates a new KSK and obtains the DNSKEY RRset of each DNS provider using their respective APIs. The new KSK is added to each provider's DNSKEY RRset and the RRset is re-signed with both the new and the old KSK. This new DNSKEY RRset is then transferred to each provider. The Zone Owner then updates the DS RRset in the parent zone to point to the new KSK, and after the necessary DS record TTL period has expired, proceeds with updating the DNSKEY RRSet to remove the old KSK.

- Zone Signing Key Rollover: In this model, each DNS provider has separate Zone Signing Keys. Each provider can choose to roll their ZSK independently by co-ordinating with the Zone Owner. Provider A would generate a new ZSK and communicate their intent to perform a rollover (note that Provider A cannot immediately insert this new ZSK into their DNSKEY RRset because the RRset has to be signed by the Zone Owner). The Zone Owner obtains the new ZSK from Provider A. It then obtains the current DNSKEY RRset from each provider (including Provider A), inserts the new ZSK into each DNSKEY RRset, re-signs the DNSKEY RRset, and sends it back to each provider for deployment via their respective key management APIs. Once the necessary time period is elapsed (i.e. all zone data has been re-signed by the new ZSK and propagated to all authoritative servers for the zone, plus the maximum zone TTL value of any of the data in the zone signed by the old ZSK), Provider A and the zone owner can initiate the next phase of removing the old ZSK.

6.2. Model 2: Unique KSK and ZSK per provider

- Key Signing Key Rollover: In Model 2, each managed DNS provider has their own KSK. A KSK roll for provider A does not require any change in the DNSKEY RRset of provider B, but does require co-ordination with the Zone Owner in order to get the DS record set in the parent zone updated. The KSK roll starts with Provider A generating a new KSK and including it in their DNSKEY RRSet. The DNSKey RRset would then be signed by both the new and old KSK. The new KSK is communicated to the Zone Owner, after which the Zone Owner updates the DS RRset to replace the DS record for the old KSK with a DS record for the new KSK. After the necessary DS RRset TTL period has elapsed, the old KSK can be removed from provider A's DNSKEY RRset.

- Zone Signing Key Rollover: In Model 2, each managed DNS provider has their own ZSK. The ZSK roll for provider A would start with them generating new ZSK and including it in their DNSKEY RRset and re-signing the new DNSKEY RRset with their KSK. The new ZSK of provider A would then be communicated to the Zone Owner, who will initiate the process of importing this ZSK into the DNSKEY RRsets of the other providers, using their respective APIs. Once the necessary Pre-Publish key rollover time periods have elapsed, provider A and the Zone Owner can initiate the process of removing the old ZSK from the DNSKEY RRset of all providers.

7. Using Combined Signing Keys

A Combined Signing Key (CSK), is one in which the same key serves the purpose of being both the secure entry point (SEP) key for the zone, and also for signing all the zone data including the DNSKEY RRset (i.e. there is no KSK/ZSK split).

Model 1 is not compatible with CSKs because the zone owner would then hold the sole signing key, and providers would not be able to sign their own zone data.

Model 2 can accommodate CSKs without issue. In this case, any or all the providers could employ a CSK. The DS record in the parent zone would reference the provider's CSK instead of KSK, and the public CSK will need to be imported into the DNSKEY RRsets of all the other providers. A CSK key rollover for such a provider would involve the following: The provider generates a new CSK, installs the new CSK into the DNSKEY RRset, and signs it with both the old and new CSK. The new CSK is communicated to the Zone Owner. The Zone Owner exports this CSK into the other provider's DNSKEY RRsets and replaces the DS record referencing the old CSK with one referencing the new one in the parent DS RRset. Once all the zone data has been re-signed with the new CSK, the old CSK is removed from the DNSKEY RRset, and the latter is re-signed with only the new CSK. Finally, the old CSK is removed from the DNSKEY RRsets of the other providers.

8. Use of CDS and CDNSKEY

CDS and CDNSKEY records [RFC7344] [RFC8078] are used to facilitate automated updates of DNSSEC secure entry point keys between parent and child zones. Multi-signer DNSSEC configurations can support this too. In Model 1, CDS/CDNSKEY changes are centralized at the zone owner. However, the zone owner will still need to push down updated signed CDNS/DNSKEY RRsets to the providers via the key management mechanism. In Model 2, the key management mechanism needs to support

cross importation of the CDS/CDNSKEY records, so that a common view

of the RRset can be constructed at each provider, and is visible to the parent zone attempting to update the DS RRset.

9. Key Management Mechanism Requirements

Managed DNS providers often have their own proprietary zone configuration and data management APIs, typically utilizing HTTPS/ REST interfaces. So, rather than outlining a new API for key management here, we describe the specific functions that the provider API needs to support in order to enable the multi-signer models. The Zone owner is expected to use these API functions to perform key management tasks. Other mechanisms that can offer these functions, if supported by the providers, include the DNS UPDATE protocol [RFC2136] and EPP [RFC5731].

- The API must offer a way to query the current DNSKEY RRset of the provider

- For model 1, the API must offer a way to import a signed DNSKEY RRset and replace the current one at the provider. Additionally, if CDS/CDNSKEY is supported, the API must also offer a way to import a signed CDS/CDNSKEY RRset.

- For model 2, the API must offer a way to import a DNSKEY record from an external provider into the current DNSKEY RRset. Additionally, if CDS/CDNSKEY is supported, the API must offer a mechanism to import individual CDS/CDNSKEY records from an external provider.

In model 2, once initially bootstrapped with each others zone signing keys via these API mechanisms, providers could, if desired, periodically query each others DNSKEY RRsets and automatically import or withdraw ZSKs in the keyset as key rollover events happen.

10. DNS Response Size Considerations

The Multi-Signer models described in this document result in larger DNSKEY RRsets, so the DNSKEY response size will be larger. The actual size depends on multiple factors: DNSKEY algorithm and keysize

choices, the number of providers, whether additional keys are pre-published, how many simultaneous key rollovers are in progress etc. Newer elliptic curve algorithms produce keys small enough that the responses will typically be far below the common Internet path MTU, and thus operational concerns related to IP fragmentation or truncation and TCP fallback are unlikely to be encountered. In any case, DNS operators need to ensure that they can emit and process large DNS responses when necessary, and a future migration to

alternative transports like DNS over TLS or HTTPS may make this topic moot.

11. IANA Considerations

This document includes no request to IANA.

12. Security Considerations

The Zone key import APIs required by these models need to be strongly authenticated to prevent tampering of key material by malicious third parties. Many providers today offer REST/HTTPS APIs that utilize a number of authentication mechanisms (username/password, API keys etc). If DNS protocol mechanisms like UPDATE are being used for key insertion and deletion, they should similarly be strongly authenticated, e.g. by employing Transaction Signatures (TSIG) [RFC2845].

13. Acknowledgments

The initial version of this document benefited from discussions with and review from Duane Wessels. Additional helpful comments were provided by Steve Crocker, Ulrich Wisser, Tony Finch, Olafur Gudmundsson, and Matthijs Mekking.

14. References

14.1. Normative References

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997,

<https://www.rfc-editor.org/info/rfc2136>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <https://www.rfc-editor.org/info/rfc2845>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <https://www.rfc-editor.org/info/rfc5155>.

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <https://www.rfc-editor.org/info/rfc5731>.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, DOI 10.17487/RFC6781, December 2012, <https://www.rfc-editor.org/info/rfc6781>.

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <https://www.rfc-editor.org/info/rfc7344>.

[RFC8078] Gudmundsson, O. and P. Wouters, "Managing DS Records from the Parent via CDS/CDNSKEY", RFC 8078, DOI 10.17487/RFC8078, March 2017, <https://www.rfc-editor.org/info/rfc8078>.

[RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <https://www.rfc-editor.org/info/rfc8198>.

14.2. Informative References

[BLACKLIES] Valsorda, F. and O. Gudmundsson, "Compact DNSSEC Denial of Existence or Black Lies", <https://tools.ietf.org/html/draft-valsorda-dnsop-black-lies>.

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <https://www.rfc-editor.org/info/rfc5936>.

[RFC7129] Gieben, R. and W. Mekking, "Authenticated Denial of Existence in the DNS", RFC 7129, DOI 10.17487/RFC7129, February 2014, <https://www.rfc-editor.org/info/rfc7129>.

Authors' Addresses

Shumon Huque
Salesforce

Email: shuque@gmail.com

Pallavi Aras
Salesforce

Email: paras@salesforce.com

John Dickinson
Sinodun

Email: jad@sinodun.com

Jan Vcelak
NS1

Email: jvcelak@ns1.com

David Blacka
Verisign

Email: davidb@verisign.com