DNSSD
Internet-Draft
Obsoletes: 2931 (if approved)
Intended status: Standards Track
Expires: 3 September 2026

D. Eastlake
Independent
J. Stenstam
Swedish Internet Foundation
M. Andrews
ISC

2 March 2026

Domain Name System (DNS) Public Key Based Request and Transaction Authentication (SIGZERO, SIG(0)) draft-eastlake-dnssd-rfc2931bis-sigzero-01

Abstract

This document specifies use of the SIGZERO and SIG(0) Domain Name System (DNS) Resource Records (RRs) to provide public key based authentication of DNS requests and transactions. This document obsoletes RFC 2931.

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 3 September 2026.

Copyright Notice

Table of Contents

1. Introduction

[[[META COMMENTS: Comments within triple square brackets like this are discussion points or alternatives/options for the content of this draft. They are NOT text to be included in the final document.]]]

This document specifies use of the Domain Name System (DNS) SIGZERO and SIG(0) Resource Records (RRs) to provide public key based authenticate of DNS requests and transactions. SIGZERO is patterned after the TSIG [RFC8945] and RRSIG [RFC4034] RRs. Use of SIGZERO is RECOMMENDED for this purpose.

As discussed below, use of the SIG RR for this purpose is NOT RECOMMENDED. When a SIG RR is so used, it has a "type covered" field value of zero, and is called a "SIG(0)" RR.

This document obsoletes [RFC2931].

2. Terminology

The term "SIG Zero" is used in this document to refer to both the SIGZERO RR and the SIG(0) RR.

General familiarity with DNS terminology [RFC9499] is assumed.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. SIG Zero RR Design Rationale and Use

The SIGZERO and SIG(0) (SIG Zero) Meta-RRs provide authentication and integrity protection for DNS transactions and requests that is not provided by the DNSSEC security specified in [RFC4034] and [RFC4035]. The authenticated data origin services of DNSSEC security provide either protected data resource records (RRs) or authenticatable denial of their existence. Those services provide no protection for the following:

* glue records,
* DNS requests,
* the DNS message headers on requests or responses, and
* the overall integrity of a transaction, that is to say, that the response was issued in response to the request sent and neither was tampered with in transit.

As Meta-RRs, SIGZERO and SIG(0) are not stored in zones. They accompany DNS messages in the Additional Information section of the message. (See Section 7.1 for a discussion of forwarding them in the case of a recursive server.)

Transaction authentication can provide cryptographic assurance to a resolver that it is at least getting the DNS response message from the server and that the received message is in response to the request it sent. This is accomplished by adding either a TSIG RR [RFC8945] or, as specified herein, a SIG Zero RR, at the end of the additional information section of the response which authenticates the concatenation of the corresponding resolver request and the server's response.

Requests can also be authenticated by including a SIG Zero RR at the end of the request's additional information section. The method of signing requests as specified herein is primarily intended for authenticating dynamic update requests [RFC3007], use in the Service Registration Protocol (SRP) for DNS-Based Service Discovery [RFC9665], TKEY requests [rfc2930bis], or other requests specified in the future that require authentication.

The private keys used in public key based request security belong to the host composing the DNS request message or other entity composing the request or to a zone to be affected by the request. The corresponding public key(s) can be stored in and retrieved from the DNS for verification as KEY RRs with a protocol byte of 3 or 255 (ANY).

4. Differences Between TSIG, SIGZERO, and SIG(0)

A TSIG [RFC8945] RR can also be used for request and transaction authentication. There are significant differences between TSIG and SIG Zero.

Because TSIG involves secret keys available at both the requester and server the presence of such a key can imply that it is likely that the other party understands TSIG and has the same key installed. Furthermore, TSIG uses keyed hash authentication codes which are relatively inexpensive to compute. Thus, it is common to authenticate DNS requests with TSIG and to authenticate DNS transactaions with TSIG if the corresponding request is authenticated.

SIGZERO and SIG(0) on the other hand, uses public key authentication, where the public keys can be stored in DNS as KEY RRs and a private key is held by the signer. Existence of such a KEY RR does not necessarily imply that SIGZERO or SIG(0) is implemented or enabled.

In addition, they involve relatively expensive public key cryptographic operations and their verification involves obtaining and verifying the corresponding KEY which can itself be an expensive operation. Indeed, a policy of using SIGZERO or SIG(0) on all requests and verifying it before responding would, for some configurations, lead to a deadly embrace with the attempt to obtain and verify the KEY needed to authenticate the request resulting in additional requests accompanied by a SIGZERO or SIG(0) leading to further requests accompanied by a SIGZERO or SIG(0), etc. Furthermore, omitting these RRs when not required on requests halves the number of public key operations required by the transaction.

For these reasons, SIG Zero RRs SHOULD only be used on requests when necessary to authenticate that the requester has some required privilege or identity. SIG Zero on transactions is defined in such a way as to not require a SIG Zero on the corresponding request and still provide transaction protection.

Which SIG Zeros are required to be authenticated by a server or requester should be a local configuration option.

4.1. Why SIGZERO Is Prefered Over SIG(0)

The SIGZERO RR was designed for this application and its use is RECOMMENDED. SIG(0) is an alternative use of the older SIG RR originally specified in [RFC2535] and is NOT RECOMMENDED. Four reasons for this preference are given below.

1. SIG(0) does not provide a convenient way to include an extended error code in a response reporting on the SIG Zero operation. SIGZERO has an Error field for this purpose.

2. There is no convenient way to extend SIG(0) while SIGZERO has an Other Data field for this purpose.

3. SIG(0) does not provide a convenient way to forward the original DNS request message ID through a recursive server but this value is needed to validate a SIG(0) or SIGZERO signature. SIGZERO has a reserved field for this value (as does the TSIG RR). See Section 7.1.

4. The SIG RR was used as a data RR for signatures and its RRTYPE is in the range normally used for data RRs. That, if and only if a SIG RR’s Type Covered field is zero, it is actually a Meta-RR, might confuse a defective DNS implemenetation .

4.2. Multiple TSIGs, SIGZEROs, and/or SIG(0)s

A request or response may contain any one of the following:

* one TSIG or

* one SIG(0) or

* one or more SIGZERO RRs.

It MUST NOT have both a TSIG and a SIG(0) and MUST NOT have more than one TSIG or more than one SIG(0). It MUST NOT have a TSIG or SIG(0) along with a SIGZERO. A request with these prohibited combinations MUST be rejected with FORMERR and a response with such a combination is ignored.

[[[This section preserves the current restrictions on multiple TSIGs and/or SIG(0)s. Should this be further liberalized?]]]

5. The SIG Zero Resource Records

The format of the SIGZERO and SIG(0) RRs are specified in the subsections below. All multi-octet integers in these RRs are sent in network byte order (see Section 2.3.2 of [RFC1035]).

5.1. SIGZERO RR Format

The fields of the SIGZERO Meta-RR are described below.

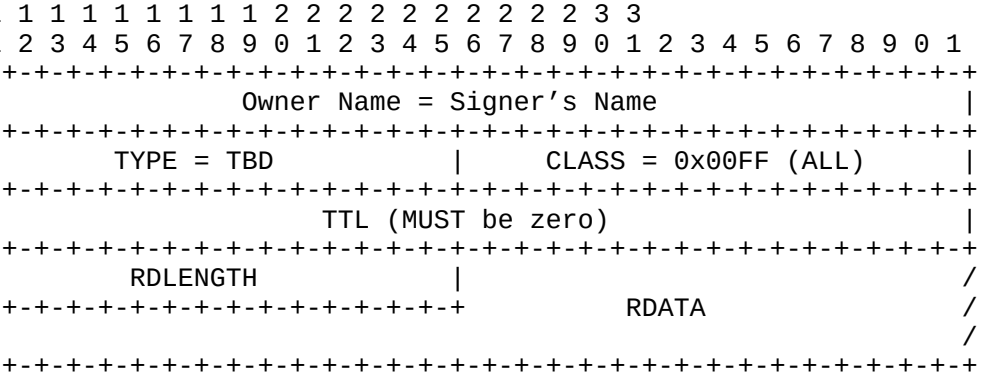

Figure 1: SIGZERO RR Format

NAME: The owner name of the KEY RR holding the public key that

should verify the signature. If there is no such KEY RR owner name then, to avoid wasting bytes, this should be the name of the root, that is, a single zero byte.

TYPE: This MUST be TBD [248 suggested] for SIGZERO

CLASS: This MUST be 255 (ANY, 0x00FF).

TTL: The TTL field MUST be zero and is ignored on receipt. The zero value minimizes the risk that a DNS implementation that does not understand SIGZERO will cache the RR.

RDLENGTH: An unsigned integer giving the length of the RDATA.

RDATA The RDATA for a SIGZERO RR consists of a number of fields as described below and shown in Figure 2.

(Most of these fields are similar or identical to the field in the TSIG [RFC8945] or RRSIG [RFC4034] RR with same field name.)

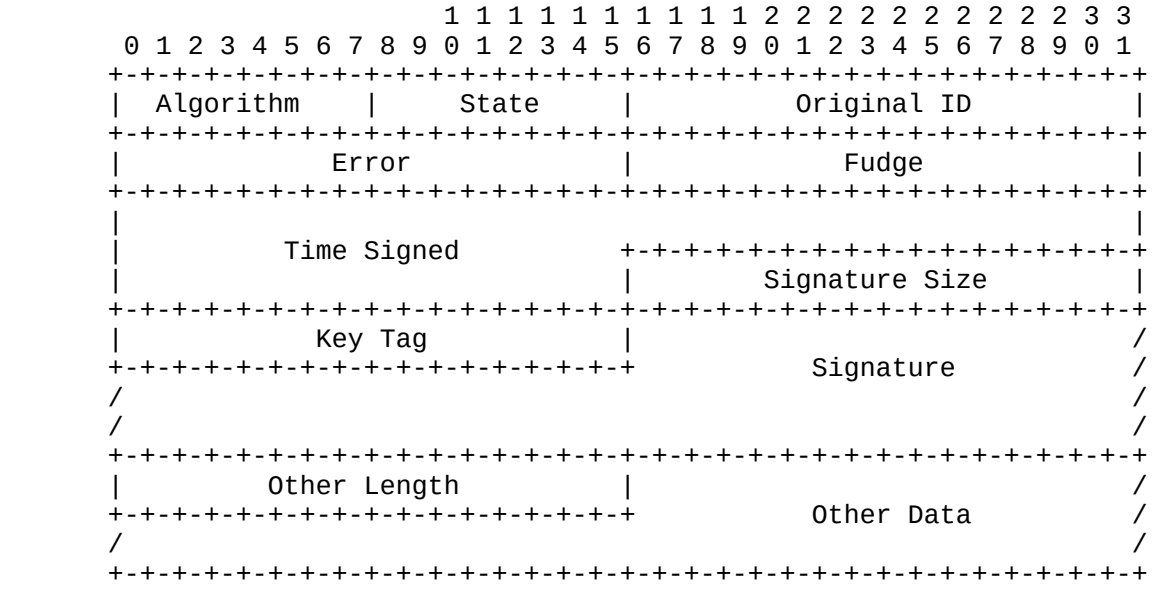

Figure 2: SIGZERO RR Format

Algorithm: The signature algorithm used. Values in this field have the same meaning as they do in the RRSIG RR [RFC4034] and use the same IANA registry [IANAalg]

State: [[[Intended as a possible location for the "key state"

suggest in draft-berra-dnsop-keystate [keystate]. If not used for that then this field would be specified as "MUST be sent as zero and ignored on receipt." It could be called "Z" or "Flags".]]]

Original ID: An unsigned 16-bit integer holding the DNS message ID of the original request message. For a SIGZERO RR added to a DNS message being originated, it is set equal to the DNS message ID unless that SIGZERO RR is being forwarded as discussed in Section 7.1 in which case the Original ID field is forwarded as received.

Error: In responses, an unsigned 16-bit integer containing the extended RCODE covering SIGZERO processing. In requests, this MUST be zero and is ignored on receipt.

Fudge: An unsigned 16-bit integer specifying the allowed time difference in seconds permitted from the Time Signed field.

Time Signed: An unsigned 48-bit integer containing the time the message was signed as seconds since 00:00 on 1970-01-01 UTC, ignoring leap seconds.

Signature Size: An unsigned integer giving the size of the Signature field in bytes.

Key Tag: The Key Tag field contains the key tag value of the public key that is to be used to validate this signature. In the case of multiple possible KEY RRs, the Key Tag usually allows a heuristic reduction the number of KEY RRs to try. Appendix B of [RFC4034] specifies how to calculate Key Tag values.

Signature: The Signature field contains the cryptographic signature that covers the DNS request or transaction. The format of this field depends on the algorithm in use, and these formats are described in separate companion documents referenced from the IANA registry [IANAalg].

Other Length: An unsigned 16-bit integer specifying the length of the Other Data field in octets.

Other Data: Additional data relevant to the TSIG record to be specified in later extentions.

The Time Signed and the Fudge form a time bracket, that is, the Time Signed plus/minus the Fudge, for the purpose of limiting replay attacks. Signatures received outside that bracket are considered invalid. In IP networks, the value of Fudge should normally not be more than 5 minutes.

[[[Should it be permissible to omit Other Length if it is zero? Should the Other Length / Other Data construct copied from TSIG and TKEY be replaced by TLVs, which would be more flexible?]]]

5.2. The SIG(0) RR Format

The structure of the SIG(0) resource record (RR) is the same as the structure of the RRSIG RR in [RFC4034] Section 4 except as provided below.

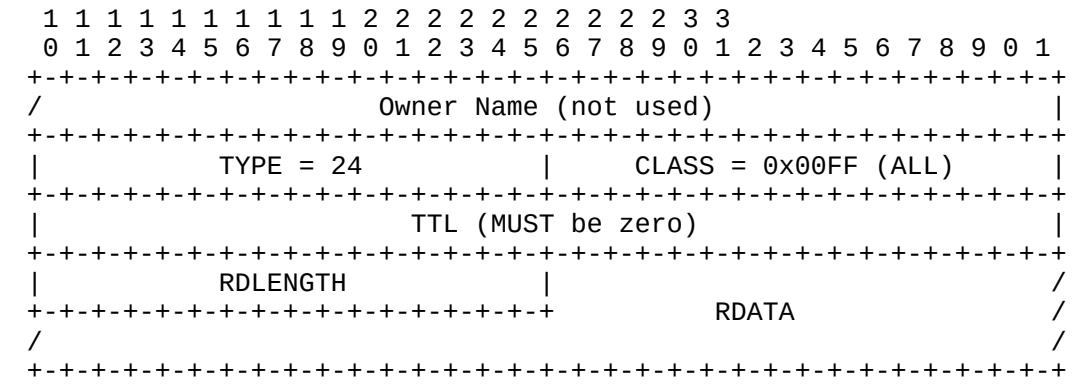

Figure 3: SIG(0) RR Format

The type of the SIG RR is 24. The structure of a SIG(0) RDATA is shown in Figure 4.

For SIG(0) RRs, the owner name, CLASS, TTL, and original TTL, and Labels fields are meaningless. To conserve space, the owner name SHOULD be root (a single zero octet). The TTL field MUST be zero. The Labels field MUST be zero. The CLASS field MUST be 255 (ANY), These fields are ignored on receipt. A TTL of zero decreases the risk that a DNS implementation that does not understand SIG(0) will cache such an RR. The RDATA for a SIG(0) RR consists of a number of fields as described below.

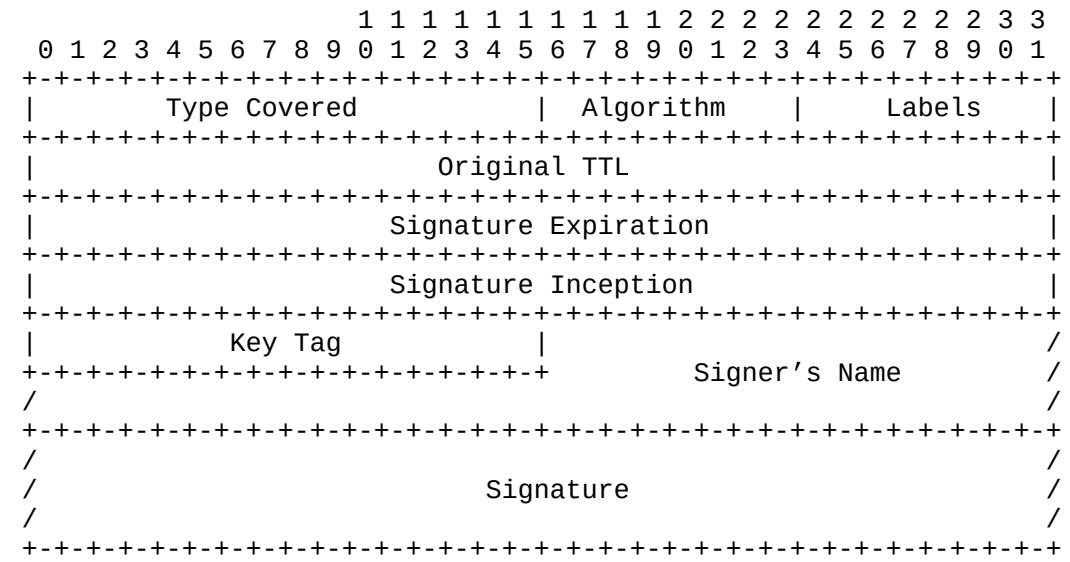

Figure 4: SIG(0) RDATA Format from [RFC4034]

The Inception and Expiration times in SIG(0)s are for the purpose of limiting replay attacks. They form a time bracket and messages received outside that bracket are ignored. In IP networks, this time bracket SHOULD NOT normally extend further than 5 minutes into the past and 5 minutes into the future.

For transaction SIG(0)s, there MUST be a KEY RR associated with the signer name with the public key corresponding to the private key used to calculate the signature. For general transaction authentication and integrity, the signer field is a name of the originating host; for example, the host domain name used may be the inverse IP address mapping name for an IP address of the host if the relevant KEY is stored there. For SIG(0)s on requests requiring authentication, the signer relates to the authority being requested such as authority to update a zone.

When SIG(0) authentication on a response is desired, the SIG(0) MUST be considered the highest priority of any additional information for inclusion in the response. If the SIG(0) RR cannot be added without causing the message to be truncated, the server MUST alter the response so that a SIG(0) can be included, set the TC bit, and return RCODE 0 (NOERROR). The client should at this point retry the request using an available transport that does not have UDP packet size restrictions.

> | Both SIG(0) DNS transaction security and DNSSEC data security
> | originally used the SIG (type = 24) and KEY (type = 25) RRs.
> | DNSSEC was changed to use the RRSIG (type = 46) and DNSKEY
> | (type = 48) RRs [RFC4034]; however, SIG(0), continues to use
> | the SIG and KEY RRs and SIGZERO makes use of the KEY RR.

6. Calculating Request and Transaction SIG Zero RRs

This section specifies the data over which request and transaction SIG Zeros are calculated. In the case of more than one SIGZERO RR, there is no significance to their order and a SIGZERO RR does not sign any other SIGZERO RR in that DNS message.

6.1. Calculating Request SIGZERO RRs

A DNS request may be optionally signed by including one or more SIGZERO RRs at the end of the request additional information section. These RRs are calculated for and sign the "data" consisting of the following:

1. The SIGZERO RR with the signature field set to zero.
2. The DNS request message, including the DNS header with the Message ID replaced by the Original ID field of the SIGZERO, but not any earlier headers such as UDP/IP headers, and before any SIG Zero RRs have been added so that, for example, the request RR counts have not yet been adjusted for SIG Zero inclusion.

That is

```
data = (SIGZERO RR) |
       (message with Message ID replaced and without SIG Zero RRs)
```

where "|" is concatenation and RR is a SIGZERO RR with the signature field set to zero.

6.2. Calculating Request SIG(0) RRs

A DNS request may be optionally signed by including a single SIG(0) RR) at the end of the request additional information section. This RR is calculated for and signs the "data" consisting of the following:

1. For the SIG(0), its RDATA section omitting the signature subfield itself, not just zeroing its value.
2. The DNS request message, including DNS header, but not any earlier headers such as UDP/IP headers and before the SIG(0) RR has been added so that, for example, the request RR counts have not yet been adjusted for SIG(0) inclusion.

That is

data = (SIG(0) RDATA) | (request without SIG(0))

where "|" is concatenation and RDATA is the RDATA of a SIG(0) being calculated omitting the signature itself.

6.3. Calculating Transaction SIGZERO RRs

A SIGZERO can be used to secure a transacation consisting of a response and the request that produced it. Such a transaction signature, to be included in the response, is calculated over data consisting of

1. That SIGZERO RR with the signature field set zero.
2. The DNS request message that produced this response, omitting any SIG Zeros that were present in the request and including the request's DNS header with the Message ID replaced by the Original ID field of the SIGZERO but not any preceding headers such as UDP or IP. (Request SIG Zeros are omitted to avoid possible confusion if, for example, a forwarder adds a SIGZERO to a request.)
3. The DNS response message, including DNS header but not any preceding headers such as TCP or IP and before any SIGZEROs have been added so that, for example, the response RR counts have not yet been adjusted for such inclusion.

data = (SIGZERO RR) | (request message without SIG Zeros) | (response message without SIG Zeros)

where "|" is concatenation.

Successful verification of a response SIGZERO by the requesting resolver increases confidence that the query and response were not tampered with in transit, that the response corresponds to the intended query, and that the response comes from the queried server.

6.4. Calculating a Transaction SIG(0) RR

A SIG(0) can be used to secure a transacation consisting of a response and the request that produced it. Such a transaction signature is calculated by using a "data" consisting of

1. For the SIG(0), the RDATA section omitting (not just zeroing) the signature itself.
2. The DNS request message that produced this response, including the request's DNS header and any SIG(0) that was present but not any preceding headers such as UDP or IP.

3. The DNS response message, including DNS header but not any preceding headers such as TCP or IP and before any SIG(0) has been added so that, for example, the response RR counts have not yet been adjusted for such inclusion.

```
data =
  (SIG(0) RDATA) | request message |
  (response message without SIG Zeros)
```

where "|" is concatenation and RDATA is the RDATA of a SIG(0) being calculated omitting the signature itself.

Successful verification of a response SIG Zero by the requesting resolver increases confidence that the query and response were not tampered with in transit, that the response corresponds to the intended query, and that the response comes from the queried server.

6.5. Handling Multipacket DNS Messages

In the case of a DNS message split into multiple packets via a stream transport (e.g. TCP, DoT, etc.), a SIG Zero on the first data packet is calculated with "data" as above but using data only from the first packet. For each subsequent packet, it is calculated as follows:

```
data =  ( (SIG(0) RDATA) or (SIGZERO RR) ) |
        (DNS payload without SIG Zeros) | previous packet
```

where "|" is concatenations, RDATA and SIGZERO RR are as above, and previous packet is the previous DNS payload including DNS header but not any preceding headers such as TCP or IP and including a SIG(0) RR if one was present but not any SIGZERO RRs.

This does not apply to fragmented UDP where the message is just signed once at the end and then fragmented.

Except where needed to authenticate an update, TKEY, or similar privileged request, servers are not required to check a request SIG Zero.

7. Processing SIG Zero RRs and Responses

If the time when a SIG Zero on a request is received is outside the interval indicated (by the Inception and Expiration Times for a SIG(0) or the Time Signed and Fudge for a SIGZERO), the BADTIME error is returned. If this applies to a response, the response is ignored.

If one or more SIG Zero RRs are at the end of the additional information section of a response, they are transaction signatures covering the response and the request that produced that response.

If a response's SIG Zero check succeeds, such a transaction authentication does NOT directly authenticate the validity of any data RRs in the message. However, it can increase confidence that they were sent by the queried server and have not been altered in transit. (Only an RRSIG RR [RFC4034] signed by the zone or a key tracing its authority to the zone or to resolver configuration can directly authenticate data RRs, depending on resolver policy.) If a resolver or server does not implement transaction and/or request SIG Zeros, it MUST ignore them without error where they are optional and treat them as failing where they are required.

If a response has multiple SIGZERO RRs and verification of some succeeds and other fail, the appropriate acion is dependent on local policy and configuration.

7.1. Considerations for Forwarding Servers

A server acting as a forwarding (recursive) server of a DNS message SHOULD check for the existence of SIG Zero RRs. If it implements SIG Zero RRs but cannot verify a SIG Zero RR, the server MUST include that RR when forwarding the message. If a SIG Zero passes all checks and verifies, the forwarding server SHOULD remove it and MUST, if possible, add a SIG Zero of its own to the destination or the next forwarder. If no transaction security is available to the destination and the message is a request, and if the corresponding response has the AD flag (see [RFC4035]) set, the forwarder MUST clear the AD flag before adding a SIG Zero to the response and returning the result to the host from which it received the request.

The transit of a DNS request between the originating client and final server through one or more forwarding servers and the similar handling of its response introduces additional potential failure and/or compromise points. Thus, when practical, requests with a SIG Zero RR should be sent directly to the final server that will verify that SIG Zero.

8. Security Considerations

Private keys used to create SIG Zero RRs are very sensitive information and all available steps should be taken to protect them on every host on which they are stored. Such hosts may need to be physically protected. If they are multi-user machines, great care should be taken so that unprivileged users have no access to private keying material.

The inclusion of the SIG Zero Inception and Expiration Time and the SIGZERO Time Signed and Fudge under the signature improves resistance to replay attacks.

9. IANA Considerations

IANA is requested to assign a SIGZERO RRTYPE number in the "Resource Record (RR) TYPEs" registry [DNSparams] in the range for meta-RRs as requested in Appendix A. [Value 248 is suggested] The resulting entry would be as follows:

TYPE	Value	Meaning	Reference
SIGZERO	TBD	Public key request or transaction signature.	[this document]

Table 1

10. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

11. Informative References

[keystate] Bergstrom, E., Fernandez, L., and J. Stenstam, "Signalling Key State Via DNS EDNS(0) OPT", work in progress, <https://datatracker.ietf.org/doc/draft-berra-dnsop-keystate/>.

[IANAalg] IANA, "DNS Security Algorithm Numbers", <https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml>.

[DNSparams] IANA, "Domain Name System (DNS) Parameters", <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml>.

[rfc2930bis] Eastlake, D. and M. Andrews, "Secret Key Agreement for DNS (TKEY Resource Record)", work in process, <https://datatracker.ietf.org/doc/draft-eastlake-dnsop-rfc2930bis-tkey/>.

[RFC2535] Eastlake 3rd, D., "Domain Name System Security Extensions", RFC 2535, DOI 10.17487/RFC2535, March 1999, <https://www.rfc-editor.org/info/rfc2535>.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, DOI 10.17487/RFC2931, September 2000, <https://www.rfc-editor.org/info/rfc2931>.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, DOI 10.17487/RFC3007, November 2000, <https://www.rfc-editor.org/info/rfc3007>.

[RFC8945] Dupont, F., Morris, S., Vixie, P., Eastlake 3rd, D., Gudmundsson, O., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", STD 93, RFC 8945, DOI 10.17487/RFC8945, November 2020, <https://www.rfc-editor.org/info/rfc8945>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/info/rfc9499>.

[RFC9665] Lemon, T. and S. Cheshire, "Service Registration Protocol for DNS-Based Service Discovery", RFC 9665, DOI 10.17487/RFC9665, June 2025, <https://www.rfc-editor.org/info/rfc9665>.

Appendix A. SIGZERO RRTYPE Assignment Application

A. Submission Date: tbd

B.1 Submission Type: [X] New RRTYPE [ ] Modification to RRTYPE
B.2 Kind of RR: [ ] Data RR [X] Meta-RR

C. Contact Information for submitter (will be publicly posted):
Name: Donald Eastlake Email Address: d3e3e3@gmail.com
International telephone number: +1-508-333-2270
Other contact handles:

D. Motivation for the new RRTYPE application.
SIG(0) has been found to have a number of deficiencies which are fixed in the specification for this new RRTYPE.

E. Description of the proposed RR type.
See draft-eastlake-dnssd-rfc2931bis-sigzero

F. What existing RRTYPE or RRTYPEs come closest to filling that need and why are they unsatisfactory?
The SIG RR (RFC 2536) with type covered of zero. Deficincies are list in draft-eastlake-dnssd-rfc2931bis-sigzero

G. What mnemonic is requested for the new RRTYPE (optional)?
SIGZERO

H. Does the requested RRTYPE make use of any existing IANA registry or require the creation of a new IANA subregistry in DNS Parameters?
Does not create any new registries and does not add entries to existing registries other than the allocation of the RRTYPE.

I. Does the proposal require/expect any changes in DNS servers/resolvers that prevent the new type from being processed as an unknown RRTYPE (see RFC 3597)?
It can be ignored but then you do not get its security benefits.

J. Comments: See draft-eastlake-dnssd-rfc2931bis-sigzero

Appendix B. Changes from [RFC2931]

1. Specify a new RR, SIGZERO, designed for the SIG Zero purpose which overcomes the deficiencies of SIG(0). RECOMMEND use of SIGZERO and make SIG(0) be NOT RECOMENDED. Add RRTYPE assigment template for SIGZERO.

2. Add section on considerations for forwarding servers. Add paragraph suggesting the avoidance of forwarding servers where practical to eliminate a potential point of failure or compromise.
3. Remove statement that TCP support for SIG(0) is OPTIONAL.
4. Editorial changes including updates to meet current Internet draft format requirements. Update references. Convert source to XMLv3.

Appendix C. Change History

RFC Editor: Please delete this section before publication.

C.1. From [RFC2931] to dnsop-00

1. Change to require KEY RRs used in connection with SIG(0) to have a protocol byte of 255 (ANY). ([RFC2931] also permits a protocol byte of 3.
2. Change implementation requirement for the TTL and CLASS field of SIG(0) RRs from SHOULD be zero and 255, respectively, to MUST have those values and are ignored on receipt.
3. Add section on considerations for forwarding servers.
4. Remove statement that TCP support for SIG(0) is OPTIONAL.
5. Specify an EDNS option to convey the original ID and return an extended error code.
6. Editorial changes including updates to meet current Internet draft format requirements. Update references. Convert source to XMLv3.

C.2. From dnsop-00 to dnsop-01

1. Add section on error return via EDNS and add IANA request for an EDNS OPT number.
2. Clarify that a SIG(0) public key can be associated with a zone or otherwise indicate authorization.
3. Add author.
4. Editorial Changes.

C.3. From dnsop-01 to dnsop-02

1. Permit multiple SIG(0)s.
2. Back out change requiring protocol 255 in SIG(0)s and again permit protocol 3 or 255.
3. Add reference to SIG(0) usage in SRP (Service Registration Protocol).
4. Editorial changes.

C.4. From dnsop-02 to dnsop-03

1. Generalize TCP references to include mentions of other stream protocols.

2. Update reference to DNSSD SRP from draft to [RFC9665].

3. Editorial changes.

C.5. From dnsop-03 to dnssd-00

1. Change to specify a new RR (SIGZERO) patterned more like TSIG. Drop EDNS(0) stuff.

2. Add paragraph suggesting the avoidance of forwarding servers where practical to eliminate a potential point of failure.

3. Add paragraph about Time Signed and Fudge for SIGZERO specifying a time interval similar to the Inception and Expiration times for SIG(0).

4. Omit request SIGZEROs from data for response SIGZERO to avoid breaking transaction security if a forwarder, for example, adds a second SIGZERO.

5. Numerous editorial changes.

C.6. From dnssd-00 to dnssd-01

1. Correct intended status to Standards Track from Informational.

2. Note that passage through a forwarding server may also be a potential point of compromise.

3. Soften assertions of "assurance" to "increase confidence".

4. Minor editorial changes.

Acknowledgements

The comments and suggestions of the following are gratefully acknowledged:

Stuart Cheshire

The comments and suggestions of the following persons were incorporated into [RFC2931], which was the previous version of this document, and are gratefully acknowledged:

Olafur Gudmundsson, Ed Lewis, Erik Nordmark, Brian Wellington.

Authors' Addresses

Donald E. Eastlake 3rd
Independent
2386 Panoramic Circle
Apopka, Florida 32703
United States of America
Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Johan Stenstam
Swedish Internet Foundation
Email: johan.stenstam@internetstiftelsen.se

M. Andrews
Internet Systems Consortium
PO Box 360
Newmarket, NH 03857
United States of America
Email: marka@isc.org