

Network Working Group
Internet-Draft
Expires: August 23, 2002

R. Arends
Nominum
M. Larson
VeriSign
D. Massey
USC/ISI
S. Rose
NIST
February 22, 2002

Resource Records for DNS Security Extensions
draft-ietf-dnsext-dnssec-records-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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/lid-abstracts.txt>.

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

This Internet-Draft will expire on August 23, 2002.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

The DNS Security Extensions (DNSSEC) introduce four resource records: the KEY, DS, SIG, and NXT resource records. This document defines the purpose and the RDATA format for each of these records. This document is part of a family of documents that describe the DNS Security Extensions (DNSSEC). The DNS Security Extensions are a collection of new resource records and protocol modifications that

Internet-Draft

DNSSEC Resource Records

February 2002

provide source authentication for the DNS. This document obsoletes [RFC 2535](#) and incorporates changes from all updates to [RFC 2535](#).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [4].

Table of Contents

1.	Introduction	4
1.1	DNSSEC Document Family	4
2.	The Key Resource Record	5
2.1	KEY RDATA Wire Format	5
2.1.1	The Flags Field	5
2.1.1.1	Explanation for Choice of Bit 7	6
2.1.2	The Protocol Octet Field	6
2.1.2.1	Explanation for a Fixed Value Protocol Octet Field	6
2.1.3	The Algorithm and Public Key Fields	6
2.2	The KEY RR Presentation Format	7
2.3	KEY RR Examples	7
2.3.1	Example 1	7
2.3.2	Example 2	7
3.	The SIG Resource Record	9
3.1	The SIG RDATA	9
3.1.1	The Type Covered Field	10
3.1.2	The Algorithm Number Field	10
3.1.3	The Labels Field	10
3.1.4	Original TTL Field	10
3.1.5	Signature Expiration and Inception Fields	10
3.1.6	The Key Tag Field	10
3.1.7	The Signer's Name Field	11
3.1.8	The Signature Field	11
3.2	The NXT RR Presentation Format (placeholder)	11
3.3	Calculating the signature	11
4.	The NXT Resource Record	13
4.1	NXT RDATA Wire Format	13
4.1.1	The Next Domain Name Field	13
4.1.2	The Type Bit Map Field	13
4.2	The NXT RR Presentation Format	14
5.	The DS Resource Record (placeholder)	15
6.	DNSSEC message bits	16
6.1	The AD and CD Header Bits	16

6.2	The DO Extended Flags Field Bit	16
7.	IANA Considerations	18
8.	Security Considerations	19
9.	Acknowledgements	20
	References	21
	Authors' Addresses	22

Arends, et al.

Expires August 23, 2002

[Page 2]

Internet-Draft

DNSSEC Resource Records

February 2002

A.	Key Tag Calculation	23
	Full Copyright Statement	24

1. Introduction

The reader is assumed to be familiar with common DNSSEC terminology as defined in [\[13\]](#) and familiar with the basic DNS concepts described in [RFC1034](#) [\[1\]](#) and [RFC1035](#) [\[2\]](#).

The DNS Security Extensions (DNSSEC) introduce four resource records: KEY, DS, SIG, and NXT resource records. This document defines the purpose of each resource record, the RDATA format, the ASCII representation, and an example of each RR type is given. Sections [2-5](#) describe the KEY, DS, SIG, and NXT records. [Section 6](#) describes the DNSSEC header bits.

1.1 DNSSEC Document Family

This document is part of a family of documents that define the DNS security extensions. The DNS security extensions (DNSSEC) are a collection of resource records and DNS protocol modifications that add source authentication to the Domain Name System (DNS). An introduction to DNSSEC and definition of common terms can be found in (RFC TBA). A description of DNS protocol modifications can be found in (RFC TBA). This document defines the DNSSEC resource records.

[2.](#) The Key Resource Record

Public keys used by the DNS infrastructure are stored in KEY resource records. A secure DNS zone will store its public key in a KEY RR and this KEY RR can be used to authenticate other RR sets in the zone. The KEY RR MAY also be used to store other types of DNS public keys, such as the keys used by SIG(0) [[10](#)] or TKEY [[9](#)]. These public keys are used to authenticate DNS messages such as a request to dynamically update a DNS zone.

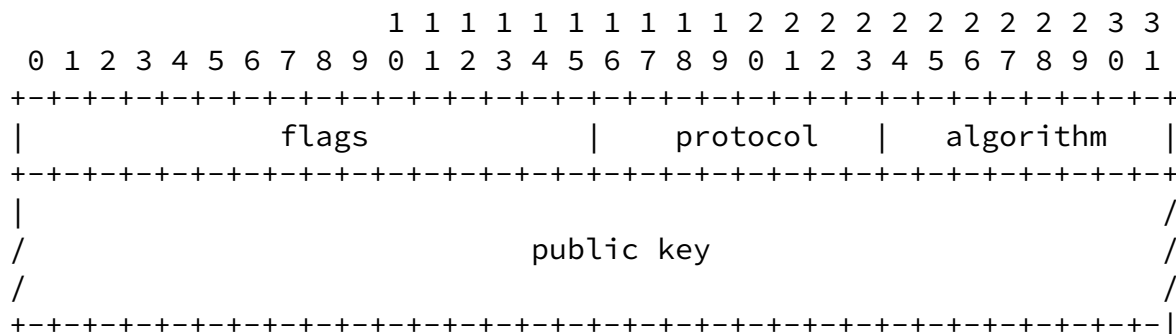
The KEY RR MUST only be used for public keys used for DNS purposes, all other uses are obsolete. The KEY RR plays an essential role in the secure processing of DNS messages and is included in various responses. The KEY RR MUST NOT be used to store certificates or public keys that do not directly relate to the DNS infrastructure. Examples of certificates and public keys that MUST NOT be stored in the KEY RR include X.509 certificates, IPSEC public keys, and SSH public keys.

The type number for the KEY RR is 25.

The KEY RR is class independent.

2.1 KEY RDATA Wire Format

The RDATA for a KEY RR consists of a 2 octet Flags Fields, a Protocol Octet, a one octet Algorithm number, and the public key.



2.1.1 The Flags Field

Bit 7 of the Flags Field is the "zone key flag". Bits 0-6 and 8-15 are reserved for future use. Bits 0-6 and 8-15 MUST be set to 0 and MUST be ignored during processing.

The zone key flag (bit 7) determines whether the KEY holds a DNS zone key. If bit 7 is 1, then the KEY record holds a DNS zone key. If bit 7 is 0, then the KEY record holds some other type of DNSSEC

infrastructure public key, such as a public key used by SIG(0) or TKEY. Resolvers MUST check the zone key flag in order to determine if the KEY record holds a DNS zone key.

2.1.1.1 Explanation for Choice of Bit 7

The choice of bit 7 as the zone key flag was made in order to provide backwards compatibility with an earlier version of the KEY record. This earlier version was defined in [6] and [15] eliminated all flags except the bit 7 zone key flag.

2.1.2 The Protocol Octet Field

The Protocol Octet value MUST be 3. Name servers and resolvers

SHOULD reject KEY records with a Protocol Octet value other than 3.

[2.1.2.1](#) Explanation for a Fixed Value Protocol Octet Field

The Protocol Octet field is included for backwards compatibility with an earlier version of the KEY record. This earlier version of the KEY record was defined in [6] and [15] restricted the possible Protocol Octet values to 3.

[2.1.3](#) The Algorithm and Public Key Fields

The Algorithm Field identifies the public key's cryptographic algorithm and determines the format of the Public Key Field.

Algorithm values are defined in separate documents. The following table shows the currently defined Algorithm formats:

VALUE	Algorithm	RFC	STATUS
0	Reserved	-	-
1	RSA/MD5	RFC 2536	NOT RECOMMENDED
2	Diffie-Hellman	RFC 2539	OPTIONAL
3	DSA	RFC 2536	MANDATORY
4	elliptic curve	-	reserved
5	RSA/SHA1	RFC 3110	MANDATORY
6-251	available for assignment	-	
252	reserved	-	indirect keys
253	private	-	domain name
254	private	-	OID
255	reserved	-	-

EDITORS NOTE: indirect keys (252), private keys 253/254 and the implication of making a key MANDATORY need further clarification. This clarification will be in the next version of this document.

[2.2](#) The KEY RR Presentation Format

A KEY RR may appear as a single line. The presentation format of the RDATA portion is as follows:

The Flag field is represented as an unsigned integer.

The Protocol Octet field is represented as the unsigned integer 3.

The Algorithm Field is represented as an unsigned integer or as mnemonic specified. The mnemonic is listed in the document defining the algorithm.

The Public Key Field is a Base 64 encoding of the Public Key Field.

[2.3](#) KEY RR Examples

[2.3.1](#) Example 1

The following KEY RR stores a DNS zone key for isi.edu.

```
isi.edu. 86400 IN KEY 256 3 5 ( AQPT0sh3WjVeRY3WqpBjtf
                                <snip of base64 encoded text>
                                xxDw==)
```

256 indicates the flags field has the zone key bit is set. 3 is the fixed Protocol Octet value. 5 indicates the public key algorithm is RSA/SHA1 [RFC 3110](#)]. The remaining text is base 64 encoding of the public key and the format of the public key is defined in [\[12\]](#).

Resolvers might use this public key to authenticate signed RR sets such as the A RR set for www.isi.edu. The authentication process used by resolvers is described in [\[14\]](#).

[2.3.2](#) Example 2

The following KEY RR stores a public key used by SIG(0)

```
ddnskey.isi.edu. 86400 IN KEY 0 3 3 ( AQPT0sh3WjVeRY3WqpBjtf
                                <snip of base64 encoded text>
                                xxDw==)
```

0 indicates the flags field does not have the zone key bit is not set. 3 is the fixed Protocol Octet value. 5 indicates the public key algorithm is DSA [\[7\]](#). The remaining text is base 64 encoding of the public key and the format of the public key is defined in [\[7\]](#).

This public key can be used to sign dynamic DNS updates for the

isi.edu zone. The process is for signing the dynamic DNS updates is described in [[11](#)].

The SIG or "signature" resource record (RR) is the fundamental way that data is authenticated in the secure Domain Name System (DNS). As such it is the heart of the security provided.

The type number for the SIG RR type is 24.

The SIG RR is class independent, but MUST have the same class as the RRset it covers. The TTL for the SIG RR SHOULD be the same as the RRset it covers.

The RDATA portion of a SIG RR is as shown below:

[illegible]

years.

A SIG RR may have an expiration time numerically less than the inception time if the expiration time is near the 32-bit wrap around point and/or the signature is long lived.

[3.1.6](#) The Key Tag Field

The "Key Tag" is a two-octet quantity that is used to efficiently select between multiple keys that may be applicable. The Key Tag value may differ depending on the key algorithm in use, as described in Appendix (A).

Arends, et al.

Expires August 23, 2002

[Page 10]

Internet-Draft

DNSSEC Resource Records

February 2002

[3.1.7](#) The Signer's Name Field

The signer's name field MUST contain the name of the zone to which the data and signature belong. The combination of signer's name, key tag, and algorithm MUST identify a zone key if the SIG is to be considered material. In a SIG(0), the signer's name MUST be the originating host of the DNS message [[10](#)].

[3.1.8](#) The Signature Field

The actual signature portion of the SIG RR binds the other RDATA fields to the RRset of the "type covered" RRs with that owner name and class.

[3.2](#) The NXT RR Presentation Format (placeholder)

This section will be here in the next revision.

[3.3](#) Calculating the signature

To generate the signature over an RRset, a data sequence is constructed as follows (where "|" is concatenation):

```
signature = sign(RDATA | RR(1) | RR(2)... )
```

```
RR(N) = name | class | type | original TTL(stored in SIG RDATA) |  
RDATA
```

To generate a signature over a DNS message (SIG(0)), a data sequence

is constructed as follows:

If the DNS message is sent via UDP:

$$\text{signature} = \text{sign}(\text{RDATA} \mid \text{full query} \mid \text{full response} - \text{SIG}(0))$$

If the DNS message is sent via TCP, the first packet's SIG(0) is calculated as above, with each additional packet (if any) calculated as follows:

$$\text{signature} = \text{sign}(\text{RDATA} \mid \text{DNS payload} - \text{SIG}(0) \mid \text{previous packet})$$

where "previous packet" is the previous DNS packet with accompanying SIG(0), but without any other headers (i.e. TCP/IP, etc.).

In all the examples,

RDATA is the wire format of all the RDATA fields in the SIG RR itself (including the canonical form of the signer's name) before but not

including the signature, and

RR(num) is the RRset with the same owner name and class and type covered as the SIG RR in canonical form.

Name is the Fully Qualified Domain Name (FQDN) in canonical form.

The canonical form for a Resource Record (RR) is the wire format of the RR. Names MUST be expanded (no name compression allowed). Name characters MUST be set to lower case. Wildcards MUST be unexpanded. The RR MUST have the original TTL.

How this data sequence is processed into the signature is algorithm dependent. These algorithm dependent formats and procedures are described in separate documents.

SIGs SHOULD NOT be generated for any "meta-type" such as ANY, AXFR, etc.

[4.](#) The NXT Resource Record

The collection of NXT or "next" resource records (RR) is used to indicate what names and RRsets [\[5\]](#) exist in a zone.

The NXT RR lists the next canonical name in the zone and lists what RR types are present for the current name of the NXT RR.

The set of NXT RRs in a zone is a chain of all authoritative names in that zone.

Glue address records MUST NOT be covered by a NXT RR.

The type number for the NXT RR is 30.

The NXT RR is class independant.

The NXT RR TTL SHOULD NOT exceed the zone minimum TTL.

[4.1](#) NXT RDATA Wire Format

The RDATA of the NXT RR is as shown below:

```

          1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               next domain name                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               type bit map                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

[4.1.1](#) The Next Domain Name Field

The "next domain name" field contains the next owner name in canonical order. Canonical order means sorted by label, highest level label first. The "next domain name" field of the NXT RR at the last name in the zone contains the zone apex name.

Glue address record names MUST NOT be covered by the "next domain name" field.

The "next domain name" field allows message compression.

[4.1.2](#) The Type Bit Map Field

The "type bit map" field format contains a single bit per RR type for RRsets with the same owner name as the NXT RR. A one bit indicates

that an RRset of that type exist for the owner name. A zero bit indicates that no RRset of that type exist for the owner name.

The first bit represents RR type zero. RR type number zero is not assigned and the corresponding bit MUST be zero. If the zero bit is one, it indicates that an unspecified format is used. This format is not used when there exist an RR type number greater than 127.

The OPT RR [8] type MUST NOT be covered by the type bit map field since it is not part of the zone data. The corresponding OPT RR type bit (40) MUST be zero.

Trailing zero octets MUST be omitted. Trailing zero octets not specified MUST be interpreted as zero octets. Glue address record types MUST NOT be covered by the type bit map field.

[4.2](#) The NXT RR Presentation Format

A NXT RR may appear as a single line. The presentation format of the RDATA portion is as follows:

The "next domain name" field is represented as a domain name.

The "type bit map" field is represented as a sequence of RR type mnemonics or as an unsigned integer.

[5](#). The DS Resource Record (placeholder)

[This section will be finalised once DS has WG consensus and is proposed standard]

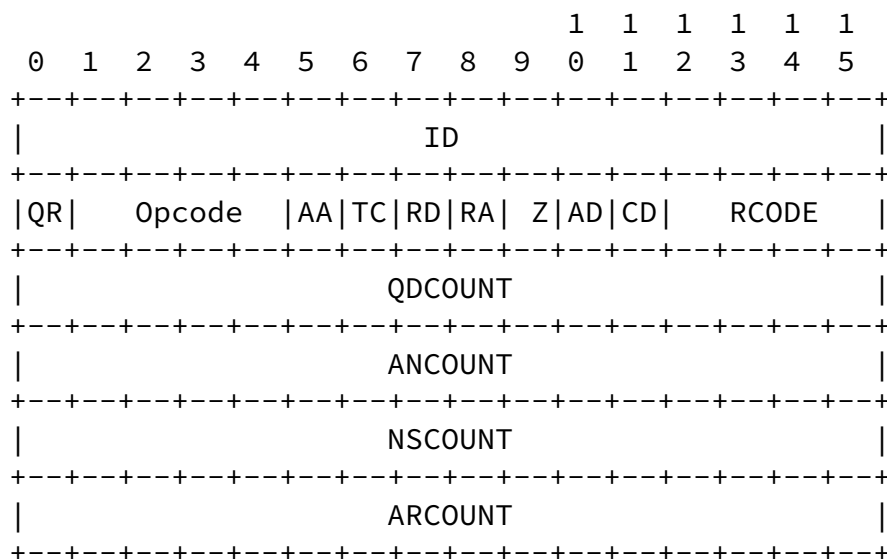
6. DNSSEC message bits

There are 3 new bits allocated for use with DNSSEC. The DO bit is used to indicate to a server that the resolver is able to accept DNSSEC security RRs (KEY SIG NXT DS). The CD and AD bits are used to indicate if non-authenticated data is accepted, and if data is authenticated.

6.1 The AD and CD Header Bits

Two bits are allocated in the header section. The CD (checking disabled) bit and the AD (authentic data) bit.

The Header contains the following fields:

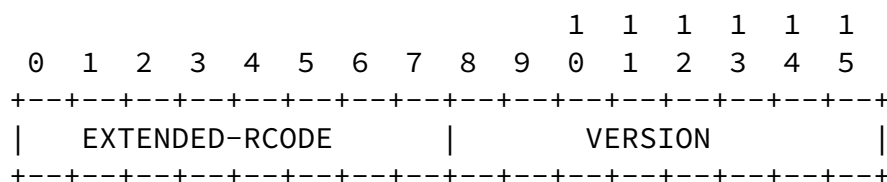


The usage of the CD and AD bits are defined in [14]

6.2 The DO Extended Flags Field Bit

The DO (DNSSEC OK) bit is allocated from the EDNS0 [8] extended flags field. In the context of the OPT RR, the DO bit is the most significant bit in the 3rd octet of the TTL field.

The TTL field of the OPT RR is defined as follows:



[7](#). IANA Considerations

This document clarifies the use of existing types and introduces no new IANA considerations.

[8](#). Security Considerations

This document describes the format of resource records used by DNS security. The threats facing DNS are described in a separate document and these records are used to help counter those threats. The records themselves introduce no new security considerations, but the protocol use of these records is described in a second document.

[9.](#) Acknowledgements

References

- [1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), November 1987.
- [2] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), November 1987.
- [3] Elz, R. and R. Bush, "Serial Number Arithmetic", [RFC 1982](#), August 1996.

- [4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [5] Elz, R. and R. Bush, "Clarifications to the DNS Specification", [RFC 2181](#), July 1997.
- [6] Eastlake, D., "Domain Name System Security Extensions", [RFC 2535](#), March 1999.
- [7] Eastlake, D., "DSA KEYS and SIGs in the Domain Name System (DNS)", [RFC 2536](#), March 1999.
- [8] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", [RFC 2671](#), August 1999.
- [9] Eastlake, D., "Secret Key Establishment for DNS (TKEY RR)", [RFC 2930](#), September 2000.
- [10] Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)", [RFC 2931](#), September 2000.
- [11] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", [RFC 3007](#), November 2000.
- [12] Eastlake, D., "RSA/SHA-1 SIGs and RSA KEYS in the Domain Name System (DNS)", [RFC 3110](#), May 2001.
- [13] Arends, R., Larson, M., Massey, D. and S. Rose, "DNSSEC Intro", February 2002.
- [14] Arends, R., Larson, M., Massey, D. and S. Rose, "DNSSEC Protocol", February 2002.
- [15] Massey, D. and S. Rose, "Limiting the Scope of the KEY Resource Record", [draft-ietf-dnsext-restrict-key-for-dnssec-01](#) (work in progress), January 2002.

2385 Bay Street
Redwood City, CA 94063
USA

E-Mail: roy.arends@nominum.com

Matt Larson
VeriSign, Inc.
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA

E-Mail: mlarson@verisign.com

Dan Massey
USC Information Sciences Institute
3811 N. Fairfax Drive
Arlington, VA 22203
USA

E-Mail: masseyd@isi.edu

Scott Rose
National Institute for Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899-3460
USA

E-Mail: scott.rose@nist.gov

[Appendix A](#). Key Tag Calculation

The key tag field in the SIG RR is just a means of more efficiently selecting the correct KEY RR to use when there is more than one KEY RR candidate available, for example, in verifying a signature. It is possible for more than one candidate key to have the same tag, in which case each must be tried until one works or all fail. The following reference implementation of how to calculate the Key Tag, for all algorithms other than algorithm 1, is in ANSI C. It is coded for clarity, not efficiency.

```
/* assumes int is at least 16 bits
   first byte of the key tag is the most significant byte of return
   value
   second byte of the key tag is the least significant byte of
   return value
   */

int keytag (
    unsigned char key[], /* the RDATA part of the KEY RR */
    unsigned int keysize, /* the RDLENGTH */
)
{
    long int    ac;      /* assumed to be 32 bits or larger */

    for ( ac = 0, i = 0; i < keysize; ++i )
        ac += (i&1) ? key[i] : key[i]<<8;
    ac += (ac>>16) & 0xFFFF;
    return ac & 0xFFFF;
}
```

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

Arends, et al.

Expires August 23, 2002

[Page 24]