

April 1999

[draft-ietf-dnsind-indirect-key-00.txt](#)

Indirect KEY RRs in the Domain Name System (DNS)

Donald E. Eastlake 3rd

Status of This Document

This draft, file name [draft-ietf-dnsind-indirect-key-00.txt](#), is intended to become a Proposed Standard RFC. Distribution of this document is unlimited. Comments should be sent to the DNSSEC mailing list <dns-security@tis.com> or to the author.

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. Internet-Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as a ``working draft'' or ``work in progress.''

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

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

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

[RFC 2535] defines a means for storing cryptographic public keys in

the Domain Name System. An additional code point is defined for the algorithm field of the KEY resource record (RR) to indicate that the key is not stored in the KEY RR but is pointed to by the KEY RR. Encodings to indicate different types of key and pointer formats are specified.

[This draft is moved from the DNSSEC WG as part of that WG's merger into me DNSIND WG. It would have been [draft-ietf-dnssec-indirect-key-02.txt](#) in the DNSSEC WG.]

Table of Contents

Status of This Document.....	1
Abstract.....	1
Table of Contents.....	2
1 . Introduction.....	3
2 . The Indirect KEY RR Algorithm.....	3
2.1 The Target Type Field.....	4
2.2 The Target Algorithm Field.....	5
2.3 The Hash Fields.....	5
3 . Performance Considerations.....	6
4 . IANA Considerations.....	6
5 . Security Considerations.....	6
References.....	7
Author's Address.....	7
Expiration and File Name.....	8

INTERNET-DRAFT

Indirect KEY RRs

1. Introduction

The Domain Name System (DNS) security extensions [[RFC 2535](#)] provide for the general storage of public keys in the domain name system via the KEY resource record (RR). These KEY RRs are used in support of DNS security and may be used to support other security protocols. KEY RRs can be associated with users, zones, and hosts or other end entities named in the DNS.

For reasons given below, it will sometimes be desirable to store a key or keys elsewhere and merely point to it from the KEY RR. Indirect key storage makes it possible to point to a key service via a URL, to have a compact pointer to a larger key or set of keys, to point to a certificate either inside DNS [[RFC 2538](#)] or outside the DNS, and where appropriate, to store a key or key set applicable to many DNS entries in some place and point to it from those entries.

However, to simplify DNSSEC implementation, this technique MUST NOT be used for KEY RRs used in for verification in DNSSEC, i.e., the value of the "protocol" field of an indirect KEY RR MUST NOT be 3.

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

2. The Indirect KEY RR Algorithm

Domain Name System (DNS) KEY Resource Record (RR) [[RFC 2535](#)] algorithm number 252 is defined as the indirect key algorithm. This algorithm MAY NOT be used for zone keys in support of DNS security. All KEYS used in DNSSEC validation MUST be stored directly in the DNS.

When the algorithm byte of a KEY RR has the value 252, the "public key" portion of the RR is formatted as follows:

```

          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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           target type           | target alg. | hash type |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| hash size |           hash (variable size)           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                                                 |
/                                                                 /
/           pointer (variable size)                             /
/                                                                 /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

[2.1](#) The Target Type Field

Target type specifies the type of the key containing data being pointed at.

Target type

0 - reserved, see [section 4](#)

1 - indicates that the pointer is a domain name from which KEY RRs [[RFC 2535](#)] should be retrieved. Name compression in the pointer field is prohibited.

2 - indicates that the pointer is a null terminated character string which is a URL [[RFC 1738](#)]. For exisiting data transfer URL

schemes, such as ftp, http, shttp, etc., the data is the same as the public key portion of a KEY RR. (New URL schemes may be defined which return multiple keys.)

- 3 - indicates that the pointer is a domain name from which CERT RRs [[RFC 2538](#)] should be retrieved. Name compression in the pointer field is prohibited.
- 4 - indicates that the pointer is a null terminated character string which is a URL [[RFC 1738](#)]. For existing data transfer URL schemes, such as ftp, http, shttp, etc., the data is the same as the entire RDATA portion of a CERT RR [[RFC 2538](#)]. (New URL schemes may be defined which return multiple such data blocks.)
- 5 - indicates that the pointer is a null terminated character string which is a URL [[RFC 1738](#)]. For existing data transfer URL schemes, such as ftp, http, shttp, etc., the data is a PKCS#1 [[RFC 2437](#)] format key. (New URL schemes may be defined which return multiple keys.)
- 6 through 255 - available for assignment, see [section 4](#).
- 256 through 511 (i.e., 256 + n) - indicate that the pointer is a null terminated character string which is a URL [[RFC 1738](#)]. For existing data transfer URL schemes, such as ftp, http, shttp, etc., the data is a certificate of the type indicated by a CERT RR [[RFC 2538](#)] certificate type of n. That is, target types 257, 258, and 259 are PKIX, SPKI, and PGP certificates and target types 509 and 510 are URL and OID private certificate types. (New URL schemes may be defined which return multiple such certificates.)
- 512 through 65534 - available for assignment, see [section 4](#).
- 65535 reserved, see [section 4](#).

[2.2](#) The Target Algorithm Field

The algorithm field is as defined in [[RFC 2535](#)]. If non-zero, it specifies the algorithm type of the target key or keys pointed. If zero, it does not specify what algorithm the target key or keys apply to.

[2.3](#) The Hash Fields

If the indirecting KEY RRset [RFC 2181, 2535] is retrieved from an appropriately secure DNS zone with a resolver implementing DNS security, then there would be a high level of confidence in the entire value of the KEY RRset including any direct keys. This may or may not be true of any indirect key pointed to. If an indirect key is embodied in a certificate or retrieved via a secure protocol such as SHTTP, it may also be secure. But an indirecting KEY RR could, for example, simply have an FTP URL pointing to a binary key stored elsewhere, the retrieval of which would not be secure.

The hash option in algorithm 252 KEY RRs provides a means of extending the security of the indirecting KEY RR to the actual key material pointed at. By including a hash in a secure indirecting RR, this secure hash can be checked against the hash of the actual keying material

Type	Hash Algorithm
----	-----
0	indicates no hash present
1	MD5 [RFC 1321]
2	SHA-1
3	RIPEMD
4-252	available, see section 4
253	private, domain name (see below)
254	private, OID (see below)
255	reserved

Codes 253 and 254 indicate that a private, proprietary, local, or experimental hash algorithm is used. For code 253, the hash field begins with a wire encoded domain name (with compression prohibited) that indicates the algorithm to use. For code 254, the hash field begins with a one byte unsigned OID length followed by a BER encoded OID which indicates the algorithm to use.

The hash size field is an unsigned octet count of the hash field size less the length of any code 253 or 254 prefix. For some hash algorithms it may be fixed by the algorithm choice but this will not always be the case. For example, hash size is used to distinguish between RIPEMD-128 (16 octets) and RIPEMD-160 (20 octets). If the

hash algorithm field is 0, the hash size MUST be zero and no hash octets are present.

The hash field itself is variable size with its length specified by the hash size field and any code 253 or 254 prefix.

[3.](#) Performance Considerations

With current public key technology, an indirect key will sometimes be shorter than the keying material it points at. In addition, there can be cases where a single indirect KEY RR points to multiple keys elsewhere. This may improve DNS performance in the retrieval of the initial KEY RR. However, an additional retrieval step then needs to be done to get the actually keying material which must be added to the overall time to get the public key.

[4.](#) IANA Considerations

IETF consensus, standards action, and similar terms in this section are as define in [[RFC 2434](#)].

KEY RR algorithm number 252 was already reserved for indirect keys in [RFC 2535](#).

An IETF standards action is required to allocate target type codes hex x0000, x0006 through x00FF, x0200 through x0FFF, and xFFFF. Codes in the range x1000 through x7FFF can be allocated by an IETF consensus. Codes x8000 through xFEFF are available on a first come first serve basis. Codes xFF00 through xFFFE are available for experimentation or private local use without allocation. Use of codes in this block may result in conflicts outside such experiment or locality.

An IETF consensus is required to allocate an indirect KEY RR hash algorithm code in the range 4-252 and a standards action is required to allocate hash algorithm code 255. Codes 253 and 254 should cover requirements for local, private, or proprietary algorithms.

[5.](#) Security Considerations

The indirecting step of using an indirect KEY RR adds complexity and additional steps where security could go wrong. If the indirect key RR was retrieved from a zone that was insecure for the resolver, you have no security. If the indirect key RR, although secure itself,

INTERNET-DRAFT

Indirect KEY RRs

point to a key which can not be securely retrieved and is not validated by a secure hash in the indirect key RR, you have no security.

References

[RFC 1034](#) - P. Mockapetris, "Domain Names - Concepts and Facilities", STD 13, November 1987.

[RFC 1035](#) - P. Mockapetris, "Domain Names - Implementation and Specifications", STD 13, November 1987.

[RFC 1321](#) - R. Rivest, "The MD5 Message-Digest Algorithm", April 1992.

[RFC 1738](#) - T. Berners-Lee, L. Masinter & M. McCahill, "Uniform Resource Locators (URL)", December 1994.

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

[RFC 2181](#) - R. Elz, R. Bush, "Clarifications to the DNS Specification", July 1997.

[RFC 2434](#) - T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", October 1998.

[RFC 2437](#) - B. Kaliski, J. Staddon, "PKCS #1: RSA Cryptography Specifications Version 2.0", October 1998.

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

[RFC 2538](#) - D. Eastlake, O. Gudmundsson, "Storing Certificates in the Domain Name System (DNS)", March 1999.

Author's Address

Donald E. Eastlake 3rd
IBM

65 shindegan Hill Road, RR #1
Carmel, NY 10512 USA

Telephone: +1-914-784-7913 (w)
 +1-914-276-2668 (h)
FAX: +1-914-784-3833 (w)
EMail: dee3@us.ibm.com

D. Eastlake 3rd

[Page 7]

INTERNET-DRAFT

Indirect KEY RRs

Expiration and File Name

This draft expires October 1999.

Its file name is [draft-ietf-dnsind-indirect-key-00.txt](#).

[D.](#) Eastlake 3rd

[Page 8]