Network Working Group
Internet-Draft
Expires: August 7, 2002

J. Schlyter
Carlstedt Research &
Technology
February 6, 2002

Storing application public keys in the DNS
draft-schlyter-appkey-02.txt

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/1id-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 7, 2002.

Abstract

This document specifies a new DNS RR type for applications to store public keys in. Experience with DNSSEC has indicated that mixing DNS keys and application keys is a bad idea in many regards. The new RR expands certain fields based on experience from early DNSSEC deployment.

Table of Contents

1. Introduction

The Domain Name System Security Extensions (DNSSEC) as described in RFC 2535 [3] specifies the KEY resource record (RR). The KEY RR is specified for use both for storing keys used by the DNSSEC infrastructure itself and for storing keys used by non-DNSSEC infrastructure applications (e.g. TLS [2], email and IPsec). The issues with combining these two uses in one RR are further discussed in a draft called "Limiting the Scope of the KEY Resource Record" [10].

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 [1].

2. Comments on the KEY RR

2.1 The flag field

The KEY RR includes a flag field that specifies key usage, what kind of entity the key is associated with and various other flags. As this kind of information often is application dependent and a common specification that covers all kinds of different flags that an application might need is hard to do, the usability of this field is questionable.

2.2 The protocol field

The protocol field in the KEY RR is only 8-bit and thus limited to 256 different protocols. As there is no way of separating different version of a specific protocol, incompatible versions of a single protocol requires multiple protocol values. A larger protocol field together with the possibility to specify a version of the protocol could solve this issue.

A problem with multiple applications storing their public keys at a single owner name and thus creating a very large RR set has been identified. A possible solution for this could be to use a generic protocol value [9] indicating that the actual protocol used is indicated in the owner name using a SRV-like encoding. Although this would indeed solve the problem with large RR sets when querying for an application key, it could also make the protocol field lose its value in practice as new applications would not require a new protocol value.

3. The APPKEY resource record

The APPKEY resource record (RR) is used to store a application public

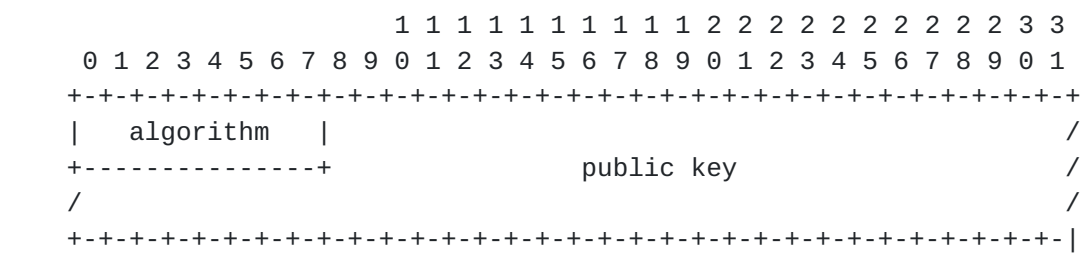
key that is associated with a Domain Name System (DNS) name.

The RR type code for the APPKEY RR is TBA.

An APPKEY RR is, like any other RR, authenticated by a SIG RR.

3.1 The APPKEY RDATA format

The RDATA for an APPKEY RR consists of an algorithm number octet and the public key itself. The format is as follows:



The meaning of the APPKEY RR owner name and algorithm number octet are described in the sections below. The format of the public key is algorithm dependent.

APPKEY RRs do not specify their validity period but their authenticating SIG RR(s) do as described in RFC 2535 [3].

3.2 Algorithm number specification

The algorithm number used is the same as defined for the KEY RR described in RFC 2535 [3].

3.3 Text representation of APPKEY RRs

The RDATA portion of an APPKEY RR has the algorithm number octet represented as unsigned integers.

The public key fields is represented in base 64 [8] and may be divided up into any number of white space separated substrings, down to single base 64 digits, which are concatenated to obtain the full public key. These substrings can span lines using the standard parenthesis notation. Note that although the public key field may have internal sub-fields, these do not appear in the master file representation.

3.4 Owner names for APPKEY RRs

The owner name of the APPKEY RR is defined per application and SHOULD be defined in such a way that it is possible to query for a single

application APPKEY. This can be, but is not limited to, the domain name of the host the application is running at (e.g. host.example.com) combined with a protocol identifier. A name matching the SRV RR [5] for the service (e.g. _service._protocol.host.example.com) could be a good starting point for this naming.

4. Applicability Statement

The APPKEY resource record (RR) are only intended for storage of public keys - private keys MUST NOT be stored in an APPKEY RR.

The APPKEY RR is not intended for storage of certificates and a separate certificate RR, defined in RFC 2538 [4], has been developed for that purpose.

5. Security considerations

Public keys from an APPKEY RR, SHOULD NOT be trusted unless the APPKEY was authenticated by a trusted SIG RR. Applications that do not validate the signatures by themselves are advised to use TSIG [6] or SIG(0) [7] to protect the transport between themselves and the name server doing the signature validation.

6. IANA considerations

IANA needs to allocate a RR type code for APPKEY from the standard RR type space. No other IANA services are required by this document.

References

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

[2] Dierks, T., Allen, C., Treese, W., Karlton, P., Freier, A. and P. Kocher, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[3] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

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

[5] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[6] Vixie, P., Gudmundsson, O., Eastlake, D. and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[7] Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[8] Josefsson, S., "Base Encodings", work in progress draft-josefsson-base-encoding-03, November 2001.

[9] Lewis, E., "DNS KEY Resource Record Generic Protocol Value", work in progress draft-lewis-dnsext-key-genprot-00, July 2001.

[10] Massey, D. and S. Rose, "Limiting the Scope of the KEY Resource Record", work in progress draft-ietf-dnsext-restrict-key-for-dnssec-01, January 2002.

Author's Address

Jakob Schlyter
Carlstedt Research & Technology
Stora Badhusgatan 18-20
Goteborg SE-411 21
Sweden

EMail: jakob@crt.se
URI: http://www.crt.se/~jakob/

Appendix A. Acknowledgements

The author gratefully acknowledges, in no particular order, the contributions of the following persons:

Olafur Gudmundsson

Olaf Kolkman

Edward Lewis

Dan Massey

Full Copyright Statement

Acknowledgement

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