DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP

Abstract

   OpenPGP is a message format for email (and file) encryption that
   lacks a standardized lookup mechanism to securely obtain OpenPGP
   public keys. DNS-Based Authentication of Named Entities (DANE) is a
   method for publishing public keys in DNS. This document specifies a
   DANE method for publishing and locating OpenPGP public keys in DNS
   for a specific email address using a new OPENPGPKEY DNS resource
   record. Security is provided via Secure DNS, however the OPENPGPKEY
   record is not a replacement for verification of authenticity via the
   "web of trust" or manual verification. The OPENPGPKEY record can be
   used to encrypt an email that would otherwise have to be sent
   unencrypted.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for examination, experimental implementation, and
   evaluation.

   This document defines an Experimental Protocol for the Internet
   community. This document is a product of the Internet Engineering
   Task Force (IETF). It represents the consensus of the IETF
   community. It has received public review and has been approved for
   publication by the Internet Engineering Steering Group (IESG). Not
   all documents approved by the IESG are a candidate for any level of
   Internet Standard; see Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7929.

Table of Contents

1. Introduction

   OpenPGP [RFC4880] public keys are used to encrypt or sign email
   messages and files. To encrypt an email message, or verify a
   sender's OpenPGP signature, the email client Mail User Agent (MUA) or
   the email server Mail Transfer Agent (MTA) needs to locate the
   recipient's OpenPGP public key.

   OpenPGP clients have relied on centralized "well-known" key servers
   that are accessed using the HTTP Keyserver Protocol [HKP].
   Alternatively, users need to manually browse a variety of different
   front-end websites. These key servers do not require a confirmation
   of the email address used in the User ID (UID) of the uploaded
   OpenPGP public key. Attackers can -- and have -- uploaded rogue
   public keys with other people's email addresses to these key servers.

   Once uploaded, public keys cannot be deleted. People who did not
   pre-sign a key revocation can never remove their OpenPGP public key
   from these key servers once they have lost access to their private
   key. This results in receiving encrypted email that cannot be
   decrypted.

   Therefore, these key servers are not well suited to support MUAs and
   MTAs to automatically encrypt email -- especially in the absence of
   an interactive user.

   This document describes a mechanism to associate a user's OpenPGP
   public key with their email address, using the OPENPGPKEY DNS RRtype.
   These records are published in the DNS zone of the user's email
   address. If the user loses their private key, the OPENPGPKEY DNS
   record can simply be updated or removed from the zone.

   The OPENPGPKEY data is secured using Secure DNS [RFC4035].

   The main goal of the OPENPGPKEY resource record is to stop passive
   attacks against plaintext emails. While it can also thwart some
   active attacks (such as people uploading rogue keys to key servers in
   the hopes that others will encrypt to these rogue keys), this
   resource record is not a replacement for verifying OpenPGP public
   keys via the "web of trust" signatures, or manually via a fingerprint
   verification.

1.1. Experiment Goal

   This specification is one experiment in improving access to public
   keys for end-to-end email security. There are a range of ways in
   which this can reasonably be done for OpenPGP or S/MIME, for example,
   using the DNS, or SMTP, or HTTP. Proposals for each of these have

been made with various levels of support in terms of implementation
and deployment. For each such experiment, specifications such as
this will enable experiments to be carried out that may succeed or
that may uncover technical or other impediments to large- or small-
scale deployments. The IETF encourages those implementing and
deploying such experiments to publicly document their experiences so
that future specifications in this space can benefit.

This document defines an RRtype whose use is Experimental. The goal
of the experiment is to see whether encrypted email usage will
increase if an automated discovery method is available to MTAs and
MUAs to help the end user with email encryption key management.

It is unclear if this RRtype will scale to some of the larger email
service deployments. Concerns have been raised about the size of the
OPENPGPKEY record and the size of the resulting DNS zone files. This
experiment hopefully will give the working group some insight into
whether or not this is a problem.

If the experiment is successful, it is expected that the findings of
the experiment will result in an updated document for standards track
approval.

The OPENPGPKEY RRtype somewhat resembles the generic CERT record
defined in [RFC4398]. However, the CERT record uses sub-typing with
many different types of keys and certificates. It is suspected that
its general application of very different protocols (PKIX versus
OpenPGP) has been the cause for lack of implementation and
deployment. Furthermore, the CERT record uses sub-typing, which is
now considered to be a bad idea for DNS.

1.2. Terminology

   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 [RFC2119].

   This document also makes use of standard DNSSEC and DANE terminology.
   See DNSSEC [RFC4033], [RFC4034], [RFC4035], and DANE [RFC6698] for
   these terms.

2. The OPENPGPKEY Resource Record

   The OPENPGPKEY DNS resource record (RR) is used to associate an end
   entity OpenPGP Transferable Public Key (see Section 11.1 of
   [RFC4880]) with an email address, thus forming an "OpenPGP public key
   association". A user that wishes to specify more than one OpenPGP
   key, for example, because they are transitioning to a newer stronger

key, can do so by adding multiple OPENPGPKEY records. A single
OPENPGPKEY DNS record MUST only contain one OpenPGP key.

The type value allocated for the OPENPGPKEY RR type is 61. The
OPENPGPKEY RR is class independent.

2.1. The OPENPGPKEY RDATA Component

The RDATA portion of an OPENPGPKEY resource record contains a single
value consisting of a Transferable Public Key formatted as specified
in [RFC4880].

2.1.1. The OPENPGPKEY RDATA Content

An OpenPGP Transferable Public Key can be arbitrarily large. DNS
records are limited in size. When creating OPENPGPKEY DNS records,
the OpenPGP Transferable Public Key should be filtered to only
contain appropriate and useful data. At a minimum, an OPENPGPKEY
Transferable Public Key for the user hugh@example.com should contain:

```
        o The primary key X
          o One User ID Y, which SHOULD match 'hugh@example.com'
            o Self-signature from X, binding X to Y
```

If the primary key is not encryption-capable, at least one relevant
subkey should be included, resulting in an OPENPGPKEY Transferable
Public Key containing:

```
        o The primary key X
          o One User ID Y, which SHOULD match 'hugh@example.com'
            o Self-signature from X, binding X to Y
          o Encryption-capable subkey Z
            o Self-signature from X, binding Z to X
          o (Other subkeys, if relevant)
```

The user can also elect to add a few third-party certifications,
which they believe would be helpful for validation in the traditional
"web of trust". The resulting OPENPGPKEY Transferable Public Key
would then look like:

```
        o The primary key X
          o One User ID Y, which SHOULD match 'hugh@example.com'
            o Self-signature from X, binding X to Y
            o Third-party certification from V, binding Y to X
            o (Other third-party certifications, if relevant)
          o Encryption-capable subkey Z
            o Self-signature from X, binding Z to X
          o (Other subkeys, if relevant)
```

2.1.2. Reducing the Transferable Public Key Size

   When preparing a Transferable Public Key for a specific OPENPGPKEY
   RDATA format with the goal of minimizing certificate size, a user
   would typically want to:

   o Where one User ID from the certifications matches the looked-up
      address, strip away non-matching User IDs and any associated
      certifications (self-signatures or third-party certifications).

   o Strip away all User Attribute packets and associated
      certifications.

   o Strip away all expired subkeys. The user may want to keep revoked
      subkeys if these were revoked prior to their preferred expiration
      time to ensure that correspondents know about these earlier than
      expected revocations.

   o Strip away all but the most recent self-signature for the
      remaining User IDs and subkeys.

   o Optionally strip away any uninteresting or unimportant third-party
      User ID certifications. This is a value judgment by the user that
      is difficult to automate. At the very least, expired and
      superseded third-party certifications should be stripped out. The
      user should attempt to keep the most recent and most well-
      connected certifications in the "web of trust" in their
      Transferable Public Key.

2.2. The OPENPGPKEY RDATA Wire Format

   The RDATA Wire Format consists of a single OpenPGP Transferable
   Public Key as defined in Section 11.1 of [RFC4880]. Note that this
   format is without ASCII armor or base64 encoding.

2.3. The OPENPGPKEY RDATA Presentation Format

   The RDATA Presentation Format, as visible in master files [RFC1035],
   consists of a single OpenPGP Transferable Public Key as defined in
   Section 11.1 of [RFC4880] encoded in base64 as defined in Section 4
   of [RFC4648].

3. Location of the OPENPGPKEY Record

   The DNS does not allow the use of all characters that are supported
   in the "local-part" of email addresses as defined in [RFC5322] and
   [RFC6530]. Therefore, email addresses are mapped into DNS using the
   following method:

   1. The "left-hand side" of the email address, called the "local-
       part" in both the mail message format definition [RFC5322] and in
       the specification for internationalized email [RFC6530]) is
       encoded in UTF-8 (or its subset ASCII). If the local-part is
       written in another charset, it MUST be converted to UTF-8.

   2. The local-part is first canonicalized using the following rules.
       If the local-part is unquoted, any comments and/or folding
       whitespace (CFWS) around dots (".") is removed. Any enclosing
       double quotes are removed. Any literal quoting is removed.

   3. If the local-part contains any non-ASCII characters, it SHOULD be
       normalized using the Unicode Normalization Form C from
       [Unicode90]. Recommended normalization rules can be found in
       Section 10.1 of [RFC6530].

   4. The local-part is hashed using the SHA2-256 [RFC5754] algorithm,
       with the hash truncated to 28 octets and represented in its
       hexadecimal representation, to become the left-most label in the
       prepared domain name.

   5. The string "_openpgpkey" becomes the second left-most label in
       the prepared domain name.

   6. The domain name (the "right-hand side" of the email address,
       called the "domain" in [RFC5322]) is appended to the result of
       step 2 to complete the prepared domain name.

   For example, to request an OPENPGPKEY resource record for a user
   whose email address is "hugh@example.com", an OPENPGPKEY query would
   be placed for the following QNAME: "c93f1e400f26708f98cb19d936620da35
   eec8f72e57f9eec01c1afd6._openpgpkey.example.com". The corresponding
   RR in the example.com zone might look like (key shortened for
   formatting):

   c9[..]d6._openpgpkey.example.com. IN OPENPGPKEY <base64 public key>

4. Email Address Variants and Internationalization Considerations

   Mail systems usually handle variant forms of local-parts. The most
   common variants are upper- and lowercase, often automatically
   corrected when a name is recognized as such. Other variants include
   systems that ignore "noise" characters such as dots, so that local-
   parts 'johnsmith' and 'John.Smith' would be equivalent. Many systems
   allow "extensions" such as 'john-ext' or 'mary+ext' where 'john' or
   'mary' is treated as the effective local-part, and 'ext' is passed to
   the recipient for further handling. This can complicate finding the
   OPENPGPKEY record associated with the dynamically created email
   address.

   [RFC5321] and its predecessors have always made it clear that only
   the recipient MTA is allowed to interpret the local-part of an
   address. Therefore, sending MUAs and MTAs supporting OPENPGPKEY MUST
   NOT perform any kind of mapping rules based on the email address. In
   order to improve chances of finding OPENPGP RRs for a particular
   local-part, domains that allow variant forms (such as treating local-
   parts as case-insensitive) might publish OPENPGP RRs for all variants
   of local-parts, might publish variants on first use (for example, a
   webmail provider that also controls DNS for a domain can publish
   variants as used by owner of a particular local-part) or just publish
   OPENPGP RRs for the most common variants.

   Section 3 above defines how the local-part is used to determine the
   location where one looks for an OPENPGPKEY record. Given the variety
   of local-parts seen in email, designing a good experiment for this is
   difficult, as: a) some current implementations are known to lowercase
   at least US-ASCII local-parts, b) we know from (many) other
   situations that any strategy based on guessing and making multiple
   DNS queries is not going to achieve consensus for good reasons, and
   c) the underlying issues are just hard -- see Section 10.1 of
   [RFC6530] for discussion of just some of the issues that would need
   to be tackled to fully address this problem.

   However, while this specification is not the place to try to address
   these issues with local-parts, doing so is also not required to
   determine the outcome of this experiment. If this experiment
   succeeds, then further work on email addresses with non-ASCII local-
   parts will be needed and, based on the findings from this experiment,
   that would be better than doing nothing or starting this experiment
   based on a speculative approach to what is a very complex topic.

5. Application Use of OPENPGPKEY

   The OPENPGPKEY record allows an application or service to obtain an
   OpenPGP public key and use it for verifying a digital signature or
   encrypting a message to the public key. The DNS answer MUST pass
   DNSSEC validation; if DNSSEC validation reaches any state other than
   "Secure" (as specified in [RFC4035]), the DNSSEC validation MUST be
   treated as a failure.

5.1. Obtaining an OpenPGP Key for a Specific Email Address

   If no OpenPGP public keys are known for an email address, an
   OPENPGPKEY DNS lookup MAY be performed to seek the OpenPGP public key
   that corresponds to that email address. This public key can then be
   used to verify a received signed message or can be used to send out
   an encrypted email message. An application whose attempt fails to
   retrieve a DNSSEC-verified OPENPGPKEY RR from the DNS should remember
   that failure for some time to avoid sending out a DNS request for
   each email message the application is sending out; such DNS requests
   constitute a privacy leak.

5.2. Confirming that an OpenPGP Key is Current

   Locally stored OpenPGP public keys are not automatically refreshed.
   If the owner of that key creates a new OpenPGP public key, that owner
   is unable to securely notify all users and applications that have its
   old OpenPGP public key. Applications and users can perform an
   OPENPGPKEY lookup to confirm that the locally stored OpenPGP public
   key is still the correct key to use. If the locally stored OpenPGP
   public key is different from the DNSSEC-validated OpenPGP public key
   currently published in DNS, the confirmation MUST be treated as a
   failure unless the locally stored OpenPGP key signed the newly
   published OpenPGP public key found in DNS. An application that can
   interact with the user MAY ask the user for guidance; otherwise, the
   application will have to apply local policy. For privacy reasons, an
   application MUST NOT attempt to look up an OpenPGP key from DNSSEC at
   every use of that key.

5.3. Public Key UIDs and Query Names

   An OpenPGP public key can be associated with multiple email addresses
   by specifying multiple key UIDs. The OpenPGP public key obtained
   from an OPENPGPKEY RR can be used as long as the query and resulting
   data form a proper email to the UID identity association.

   CNAMEs (see [RFC2181]) and DNAMEs (see [RFC6672]) can be followed to
   obtain an OPENPGPKEY RR, as long as the original recipient's email
   address appears as one of the OpenPGP public key UIDs. For example,

if the OPENPGPKEY RR query for hugh@example.com
(8d57[...]b7._openpgpkey.example.com) yields a CNAME to
8d57[...]b7._openpgpkey.example.net, and an OPENPGPKEY RR for
8d57[...]b7._openpgpkey.example.net exists, then this OpenPGP public
key can be used, provided one of the key UIDs contains
"hugh@example.com". This public key cannot be used if it would only
contain the key UID "hugh@example.net".

If one of the OpenPGP key UIDs contains only a single wildcard as the
left-hand side of the email address, such as "*@example.com", the
OpenPGP public key may be used for any email address within that
domain. Wildcards at other locations (e.g., "hugh@*.com") or regular
expressions in key UIDs are not allowed, and any OPENPGPKEY RR
containing these MUST be ignored.

6. OpenPGP Key Size and DNS

   Due to the expected size of the OPENPGPKEY record, applications
   SHOULD use TCP -- not UDP -- to perform queries for the OPENPGPKEY
   resource record.

   Although the reliability of the transport of large DNS resource
   records has improved in the last years, it is still recommended to
   keep the DNS records as small as possible without sacrificing the
   security properties of the public key. The algorithm type and key
   size of OpenPGP keys should not be modified to accommodate this
   section.

   OpenPGP supports various attributes that do not contribute to the
   security of a key, such as an embedded image file. It is recommended
   that these properties not be exported to OpenPGP public keyrings that
   are used to create OPENPGPKEY resource records. Some OpenPGP
   software (for example, GnuPG) supports a "minimal key export" that is
   well suited to use as OPENPGPKEY RDATA. See Appendix A.

7. Security Considerations

   DNSSEC is not an alternative for the "web of trust" or for manual
   fingerprint verification by users. DANE for OpenPGP, as specified in
   this document, is a solution aimed to ease obtaining someone's public
   key. Without manual verification of the OpenPGP key obtained via
   DANE, this retrieved key should only be used for encryption if the
   only other alternative is sending the message in plaintext. While
   this thwarts all passive attacks that simply capture and log all
   plaintext email content, it is not a security measure against active
   attacks. A user who publishes an OPENPGPKEY record in DNS still

expects senders to perform their due diligence by additional (non-DNSSEC) verification of their public key via other out-of-band methods before sending any confidential or sensitive information.

In other words, the OPENPGPKEY record MUST NOT be used to send sensitive information without additional verification or confirmation that the OpenPGP key actually belongs to the target recipient.

DNSSEC does not protect the queries from Pervasive Monitoring as defined in [RFC7258]. Since DNS queries are currently mostly unencrypted, a query to look up a target OPENPGPKEY record could reveal that a user using the (monitored) recursive DNS server is attempting to send encrypted email to a target. This information is normally protected by the MUAs and MTAs by using Transport Layer Security (TLS) encryption using STARTTLS. The DNS itself can mitigate some privacy concerns, but the user needs to select a trusted DNS server that supports these privacy-enhancing features. Recursive DNS servers can support DNS Query Name Minimalisation [RFC7816], which limits leaking the QNAME to only the recursive DNS server and the nameservers of the actual zone being queried for. Recursive DNS servers can also support TLS [RFC7858] to ensure that the path between the end user and the recursive DNS server is encrypted.

Various components could be responsible for encrypting an email message to a target recipient. It could be done by the sender's MUA or a MUA plug-in or the sender's MTA. Each of these have their own characteristics. A MUA can ask the user to make a decision before continuing. The MUA can either accept or refuse a message. The MTA must deliver the message as-is, or encrypt the message before delivering. Each of these components should attempt to encrypt an unencrypted outgoing message whenever possible.

In theory, two different local-parts could hash to the same value. This document assumes that such a hash collision has a negligible chance of happening.

Organizations that are required to be able to read everyone's encrypted email should publish the escrow key as the OPENPGPKEY record. Mail servers of such organizations MAY optionally re-encrypt the message to the individual's OpenPGP key.

7.1. MTA Behavior

An MTA could be operating in a stand-alone mode, without access to the sender's OpenPGP public keyring, or in a way where it can access the user's OpenPGP public keyring. Regardless, the MTA MUST NOT modify the user's OpenPGP keyring.

An MTA sending an email MUST NOT add the public key obtained from an
OPENPGPKEY resource record to a permanent public keyring for future
use beyond the TTL.

If the obtained public key is revoked, the MTA MUST NOT use the key
for encryption, even if that would result in sending the message in
plaintext.

If a message is already encrypted, the MTA SHOULD NOT re-encrypt the
message, even if different encryption schemes or different encryption
keys would be used.

If the DNS request for an OPENPGPKEY record returned an Indeterminate
or Bogus answer as specified in [RFC4035], the MTA MUST NOT send the
message and queue the plaintext message for encrypted delivery at a
later time. If the problem persists, the email should be returned
via the regular bounce methods.

If multiple non-revoked OPENPGPKEY resource records are found, the
MTA SHOULD pick the most secure RR based on its local policy.

7.2. MUA Behavior

If the public key for a recipient obtained from the locally stored
sender's public keyring differs from the recipient's OPENPGPKEY RR,
the MUA SHOULD halt processing the message and interact with the user
to resolve the conflict before continuing to process the message.

If the public key for a recipient obtained from the locally stored
sender's public keyring contains contradicting properties for the
same key obtained from an OPENPGPKEY RR, the MUA SHOULD NOT accept
the message for delivery.

If multiple non-revoked OPENPGPKEY resource records are found, the
MUA SHOULD pick the most secure OpenPGP public key based on its local
policy.

The MUA MAY interact with the user to resolve any conflicts between
locally stored keyrings and OPENPGPKEY RRdata.

A MUA that is encrypting a message SHOULD clearly indicate to the
user the difference between encrypting to a locally stored and
previously user-verified public key and encrypting to a public key
obtained via an OPENPGPKEY resource record that was not manually
verified by the user in the past.

7.3. Response Size

   To prevent amplification attacks, an Authoritative DNS server MAY
   wish to prevent returning OPENPGPKEY records over UDP unless the
   source IP address has been confirmed with [RFC7873]. Such servers
   MUST NOT return REFUSED, but answer the query with an empty answer
   section and the truncation flag set ("TC=1").

7.4. Email Address Information Leak

   The hashing of the local-part in this document is not a security
   feature. Publishing OPENPGPKEY records will create a list of hashes
   of valid email addresses, which could simplify obtaining a list of
   valid email addresses for a particular domain. It is desirable to
   not ease the harvesting of email addresses where possible.

   The domain name part of the email address is not used as part of the
   hash so that hashes can be used in multiple zones deployed using
   DNAME [RFC6672]. This does makes it slightly easier and cheaper to
   brute-force the SHA2-256 hashes into common and short local-parts, as
   single rainbow tables can be re-used across domains. This can be
   somewhat countered by using NextSECure version 3 (NSEC3).

   DNS zones that are signed with DNSSEC using NSEC for denial of
   existence are susceptible to zone walking, a mechanism that allows
   someone to enumerate all the OPENPGPKEY hashes in a zone. This can
   be used in combination with previously hashed common or short local-
   parts (in rainbow tables) to deduce valid email addresses. DNSSEC-
   signed zones using NSEC3 for denial of existence instead of NSEC are
   significantly harder to brute-force after performing a zone walk.

7.5. Storage of OPENPGPKEY Data

   Users may have a local key store with OpenPGP public keys. An
   application supporting the use of OPENPGPKEY DNS records MUST NOT
   modify the local key store without explicit confirmation of the user,
   as the application is unaware of the user's personal policy for
   adding, removing, or updating their local key store. An application
   MAY warn the user if an OPENPGPKEY record does not match the OpenPGP
   public key in the local key store.

   Applications that cannot interact with users, such as daemon
   processes, SHOULD store OpenPGP public keys obtained via OPENPGPKEY
   up to their DNS TTL value. This avoids repeated DNS lookups that
   third parties could monitor to determine when an email is being sent
   to a particular user.

7.6. Security of OpenPGP versus DNSSEC

   Anyone who can obtain a DNSSEC private key of a domain name via
   coercion, theft, or brute-force calculations, can replace any
   OPENPGPKEY record in that zone and all of the delegated child zones.
   Any future messages encrypted with the malicious OpenPGP key could
   then be read.

   Therefore, an OpenPGP key obtained via a DNSSEC-validated OPENPGPKEY
   record can only be trusted as much as the DNS domain can be trusted,
   and is no substitute for in-person OpenPGP key verification or
   additional OpenPGP verification via "web of trust" signatures present
   on the OpenPGP in question.

8. IANA Considerations

8.1. OPENPGPKEY RRtype

   This document uses a new DNS RR type, OPENPGPKEY, whose value 61 has
   been allocated by IANA from the "Resource Record (RR) TYPEs"
   subregistry of the "Domain Name System (DNS) Parameters" registry.

   The IANA template for OPENPGPKEY is listed in Appendix B. It was
   submitted to IANA for review on July 23, 2014 and approved on August
   12, 2014.

9. References

9.1. Normative References

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
              November 1987, <http://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,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS
              Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997,
              <http://www.rfc-editor.org/info/rfc2181>.

   [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,
              <http://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,
              <http://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,
              <http://www.rfc-editor.org/info/rfc4035>.

   [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data
              Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006,
              <http://www.rfc-editor.org/info/rfc4648>.

   [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R.
              Thayer, "OpenPGP Message Format", RFC 4880,
              DOI 10.17487/RFC4880, November 2007,
              <http://www.rfc-editor.org/info/rfc4880>.

   [RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic
              Message Syntax", RFC 5754, DOI 10.17487/RFC5754, January
              2010, <http://www.rfc-editor.org/info/rfc5754>.

9.2. Informative References

   [HKP]   Shaw, D., "The OpenPGP HTTP Keyserver Protocol (HKP)",
              Work in Progress, draft-shaw-openpgp-hkp-00, March 2003.

   [MAILBOX] Levine, J., "Encoding mailbox local-parts in the DNS",
              Work in Progress, draft-levine-dns-mailbox-01, September
              2015.

   [RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record
              (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September
              2003, <http://www.rfc-editor.org/info/rfc3597>.

   [RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely
              Publish Secure Shell (SSH) Key Fingerprints", RFC 4255,
              DOI 10.17487/RFC4255, January 2006,
              <http://www.rfc-editor.org/info/rfc4255>.

   [RFC4398] Josefsson, S., "Storing Certificates in the Domain Name
              System (DNS)", RFC 4398, DOI 10.17487/RFC4398, March 2006,
              <http://www.rfc-editor.org/info/rfc4398>.

   [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321,
              DOI 10.17487/RFC5321, October 2008,
              <http://www.rfc-editor.org/info/rfc5321>.

   [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322,
              DOI 10.17487/RFC5322, October 2008,
              <http://www.rfc-editor.org/info/rfc5322>.

   [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for
              Internationalized Email", RFC 6530, DOI 10.17487/RFC6530,
              February 2012, <http://www.rfc-editor.org/info/rfc6530>.

   [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the
              DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012,
              <http://www.rfc-editor.org/info/rfc6672>.

   [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication
              of Named Entities (DANE) Transport Layer Security (TLS)
              Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August
              2012, <http://www.rfc-editor.org/info/rfc6698>.

   [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an
              Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May
              2014, <http://www.rfc-editor.org/info/rfc7258>.

   [RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve
              Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016,
              <http://www.rfc-editor.org/info/rfc7816>.

   [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D.,
              and P. Hoffman, "Specification for DNS over Transport
              Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May
              2016, <http://www.rfc-editor.org/info/rfc7858>.

   [RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS)
              Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016,
              <http://www.rfc-editor.org/info/rfc7873>.

   [SMIME]   Hoffman, P. and J. Schlyter, "Using Secure DNS to
              Associate Certificates with Domain Names For S/MIME", Work
              in Progress, draft-ietf-dane-smime-12, July 2016.

   [Unicode90]
              The Unicode Consortium, "The Unicode Standard, Version
              9.0.0", (Mountain View, CA: The Unicode Consortium,
              2016. ISBN 978-1-936213-13-9),
              <http://www.unicode.org/versions/Unicode9.0.0/>.

Appendix A. Generating OPENPGPKEY Records

   The commonly available GnuPG software can be used to generate a
   minimum Transferable Public Key for the RRdata portion of an
   OPENPGPKEY record:

   gpg --export --export-options export-minimal,no-export-attributes \
       hugh@example.com | base64

   The --armor or -a option of the gpg command should not be used, as it
   adds additional markers around the armored key.

   When DNS software reading or signing of the zone file does not yet
   support the OPENPGPKEY RRtype, the Generic Record Syntax of [RFC3597]
   can be used to generate the RDATA. One needs to calculate the number
   of octets and the actual data in hexadecimal:

   gpg --export --export-options export-minimal,no-export-attributes \
       hugh@example.com | wc -c
   gpg --export --export-options export-minimal,no-export-attributes \
       hugh@example.com | hexdump -e \
           '"\t" /1 "%.2x"' -e '/32 "\n"'

   These values can then be used to generate a generic record (line
   break has been added for formatting):

   <SHA2-256-trunc(hugh)>._openpgpkey.example.com. IN TYPE61 \# \
       <numOctets> <keydata in hex>

   The openpgpkey command in the hash-slinger software can be used to
   generate complete OPENPGPKEY records

   ~> openpgpkey --output rfc hugh@example.com
   c9[..]d6._openpgpkey.example.com. IN OPENPGPKEY mQCNAzIG[...]

   ~> openpgpkey --output generic hugh@example.com
   c9[..]d6._openpgpkey.example.com. IN TYPE61 \# 2313 99008d03[...]

Appendix B. OPENPGPKEY IANA Template

   This is a copy of the original registration template submitted to
   IANA; the text (including the references) has not been updated.

   A. Submission Date: 23-07-2014

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

   C. Contact Information for submitter (will be publicly posted):
      Name: Paul Wouters   Email Address: pwouters@redhat.com
      International telephone number: +1-647-896-3464
      Other contact handles: paul@nohats.ca

   D. Motivation for the new RRTYPE application.

      Publishing RFC-4880 OpenPGP formatted keys in DNS with DNSSEC
      protection to faciliate automatic encryption of emails in
      defense against pervasive monitoring.

   E. Description of the proposed RR type.

   http://tools.ietf.org/html/draft-ietf-dane-openpgpkey-00#section-2

   F. What existing RRTYPE or RRTYPEs come closest to filling that need
      and why are they unsatisfactory?

      The CERT RRtype is the closest match. It unfortunately depends on
      subtyping, and its use in general is no longer recommended. It
      also has no human usable presentation format. Some usage types of
      CERT require external URI's which complicates the security model.
      This was discussed in the dane working group.

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

      OPENPGPKEY

   H. Does the requested RRTYPE make use of any existing IANA registry
      or require the creation of a new IANA subregistry in DNS
      Parameters? If so, please indicate which registry is to be used
      or created. If a new subregistry is needed, specify the
      allocation policy for it and its initial contents. Also include
      what the modification procedures will be.

      The RDATA part uses the key format specified in RFC-4880, which
      itself use
      https://www.iana.org/assignments/pgp-parameters/pgp-parameters.xhtm

   This RRcode just uses the formats specified in those registries for
   its RRdata part.

 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 [RFC3597])?

    No.

 J. Comments:

    Currently, three software implementations of
    draft-ietf-dane-openpgpkey are using a private number.

Acknowledgments

   This document is based on [RFC4255] and [SMIME] whose authors are
   Paul Hoffman, Jakob Schlyter, and W. Griffin. Olafur Gudmundsson
   provided feedback and suggested various improvements. Willem Toorop
   contributed the gpg and hexdump command options. Daniel Kahn Gillmor
   provided the text describing the OpenPGP packet formats and filtering
   options. Edwin Taylor contributed language improvements for various
   iterations of this document. Text regarding email mappings was taken
   from [MAILBOX] whose author is John Levine.

Author's Address

   Paul Wouters
   Red Hat

   Email: pwouters@redhat.com