Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

P. Wouters
Red Hat

October 21, 2013

Using DANE to Associate OTR public keys with email addresses
draft-wouters-dane-otrfp-01

Abstract

The Off-The-Record Messaging protocol (OTR) exchanges public keys in-band. This document describes how to use DANE to securely associate an Instant Message user identified by their email address with an OTR public key. This association helps to authenticate users and protect against MITM attacks.

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 http://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 April 24, 2014.

Table of Contents

1. Introduction

Off-the-Record [OTRSPEC] is an encryption and authentication method for two parties exchanging messages that works independantly of the transport layer. There are OTR implementations for Instant Message networks such as [XMPP], IRC [RFC2812], commercial IM networks, the GSM SMS network and others.

OTR offers encryption, authentication, repudiation and perfect forward secrecy. To authenticate the other party after an unauthenticated Diffie-Hellman key exchange, a "long term" identity keypair is used. It is up to both users to mutually verify each other's OTR public key. One can make an out-of-band phone call, and read out each other's public key fingerprint, assuming both parties can recognise and trust each other's voice. Another option is to use the shared secret. A third option is for both parties to ask each other a question to which only the other party knows the answer.

None of the above listed methods allow a person to pre-publish their OTR public key or finger print, or allow for a trusted third party or PKI to vouch for OTR public keys. As a result, most users feel it is too cumbersome to authenticate each other. As such, these users are not protected against MITM attacks.

This document describes a mechanism to associate a user's OTR public key with their email address, using a new DNS RRtype. This is similar to the SSHFP [RFC4255] RRType, except that this method associates keys with users, not hosts. Client implementations that support this document will be able to protect their users against MITM attacks, without requiring all users to manually verify each other's identity.

1.1. 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 OTRFP Resource Record

The OTRFP DNS resource record (RR) is used to associate an end entity OTR public key with an email address, thus forming a "OTRFP public key association".

The type value allocated for the OTRFP RR type is [TBD]. The OTRFP RR is class independent. The OTRFP RR has no special TTL requirements.

2.1. Location of the OTRFP record

Domain names are prepared for requests in the following manner.

1. The user name (the "left-hand side" of the email address, called the "local-part" in the mail message format definition [RFC2822] and the "local part" in the specification for internationalized email [RFC6530]), is encoded with Base32 [RFC4648], to become the left-most label in the prepared domain name. This does not include the "@" character that separates the left and right sides of the email address.

2. The string "_otrfp" becomes the second left-most label in the prepared domain name.

3. The domain name (the "right-hand side" of the email address, called the "domain" in RFC 2822) is appended to the result of step 2 to complete the prepared domain name.

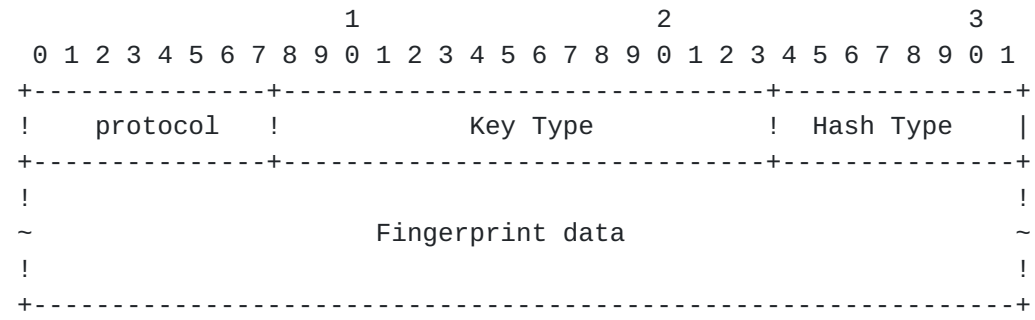
For example, to request an OTRFP resource record for a user whose address is "hugh@example.com", you would use "nb2wo2a=._otrfp.example.com" in the request. The corresponding RR in the example.com zone might look like:

```
nb2wo2a=._otrfp.example.com. IN OTRFP (
   3 0 1 5fdb8166f9089e253b90f95a91f48a8d2d2359ce )
```

Design note: Encoding the user name with Base32 allows local parts that have characters that would prevent their use in domain names. For example, a period (".") is a valid character in a local part, but would wreak havoc in a domain name. Similarly, RFC 6530 allows non-ASCII characters in local parts, and encoding a local part with non-ASCII characters with Base32 renders the name usable in the DNS.

2.2. The OTRFP RDATA Format

The RDATA for an OTRFP RR consists of an OTR protocol version, key type, hash type and the fingerprint of the user's OTR public key.



The fields have the following meaning and encoding:

Protocol:

One octet specifying the OTR protocol version. The following values are assigned:

Value	Protocol
0	reserved
1	reserved
2	version 2 (obsoleted)
3	version 3

Key Type:

Two octets specifying the OTR Key Type as defined in [draft-individual-otr]. The following values are assigned:

Value	Key Type
0	DSA

Hash Type:

One octet value specifying the hash algorithm used to compute the fingerprint data. The following values are assigned:

Value	Hash Type
0	reserved
1	SHA-1

Fingerprint data:

The actual fingerprint data in hexadecimal (see below)

3. Building the fingerprint data

OTR represents keys using multiprecision integers (also called MPIs) which are unsigned integers used to hold large integer values such as the ones used in cryptographic calculations. The OTR Public Key representation depends on the Key Type and the fingerprint is a human readable representation of the message-digest (hash) of the OTR Public Key.

3.1. Multiprecision Integers (MPI)

An MPI consists of two pieces: a two-octet scalar that is the length of the MPI in bits followed by a string of octets that contain the actual integer.

These octets form a big-endian number; a big-endian number can be made into an MPI by prefixing it with the appropriate length.

Examples (all numbers are in hexadecimal):

The string of octets [00 01 01] forms an MPI with the value 1. The string [00 09 01 FF] forms an MPI with the value of 511.

Additional rules:

The size of an MPI is ((MPI.length + 7) / 8) + 2 octets.

The length field of an MPI describes the length starting from its most significant non-zero bit. Thus, the MPI [00 02 01] is not formed correctly. It should be [00 01 01].

Unused bits of an MPI MUST be zero.

Also note that when an MPI is encrypted, the length refers to the plaintext MPI. It may be ill-formed in its ciphertext. OTR does not use encrypted MPIs.

3.2. OTR public key representation

An OTR Public Key is representated using a concatenation of its two octet Key Type, followed by the MPI typed components that make up a key. The number of components is dependant on the Key Type used.

A fingerprint is a hexadecimal representation of the message-digest (hash) of an OTR Public Key. Currently, the only supported hash is SHA-1.

There is an exception for backwards compatibility: if the OTR Key Type is 0x0000 (DSA), then the two leading 0x00 octets are omitted from the data to be hashed.

This encoding assures that, assuming the hash function itself has no useful collisions, and DSA keys have a length of less than 524281 bits (500 times larger than most DSA keys), no two public keys will have the same fingerprint.

3.3. Calculating a DSA fingerprint

For OTR Key Type 0x0000 (DSA), the public key is represented by its Key Type (0x0000) concatenated with p(MPI) q(MPI) g(MPI) y(MPI)

This representation is hashed by the desired hashing function and represented in hexadecimal to create the fingerprint data to be included in the OTRFP RDATA section.

4. IANA Considerations

4.1. OTRFP RRtype

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

5. Security Considerations

5.1. Email address information leak

DNS zones that are signed with DNSSEC using NSEC for denial of existence are susceptible to zone-walking, a mechanism that allow someone to enumerate all the names in the zone. Someone who wanted to collect email addresses from a zone that uses OTRFP might use such a mechanism. DNSSEC-signed zones using NSEC3 for denial of existence are significantly less susceptible to zone-walking. Someone could still attempt a dictionary attack on the zone to find OTRFP records, just as they can use dictionary attacks on an SMTP server to see which addresses are valid.

5.2. UI presentation

Client treatment of any information included in the OTRFP record is a matter of local policy. Clients are strongly encouraged to not visually equate a DANE verified fingerprint with a human-verified fingerprint. Padlock icons are strongly discouraged as the verification state of a public key is not a binary selection.

5.3. DNSSEC required

If an OTRFP resource record is received without DNSSEC protection, it MUST be ignored. If the DNS request returned an "indeterminate" or "bogus" answer, the user MUST be warned that they might be under attack. Bogus data MUST NOT be used.

6. Reference example

Given a textual representation of a DSA private key for hugh@example.com of:

```
 (dsa
  (p #0085CAB74C71F78ACBAF92E3959E772050E8332D1262CF7989D1ACDFBB545E
      16E757DBAAD3047E306E250C69CA4347CC7347F68070DE46B8EC4C4B41579F
      1D57F00E76EBFDD7EF5494D6E726428BE17D7E4BEFE0ECE6BA661E7BE799B6
      E5BF5EF8BB02CD1466A06114EF517CCBC21D68CC769F5A15D4FC473BA27482
      AC4F42C667#)
  (q #0086CBA0573319CFA3D3EBD8225651E58B316B22F5#)
  (g #2CE973832A3B23A9E8E5BC324E16A9445C0EBB73C03ACBBE5EEC6504F640DB
```

```
      A49C97A42282271BA6127F848EC70F1A4AB784BE0A712081DF721452C87EE6
      9B5C4FA963E933C2E592AF9631C3FDF94A85593A1BF6170889DBB8DD55A0A2
      BB19874EBDA2D96929A541576D7800BFEB467D6F991E437883F8BC7D6A3654
      5C04BDFC#)
   (y #30CCBADF74E0313E1E6274DB8CC08FA6DC5EB6FA4729BB573034D75EB564CB
      1F3DBECA70DF6A7337BD2A9EFA63986C2F64B4D4B32CFDB9F3BF6719DC4A98
      43E60319236D8EB936FFE845C5A6B856557F0E9A4CA769041FBA92CA2CCE88
      E2E3441650437FF5FB315F4AFF5CA43BFF3539B520297EC1C5BAF3E437F829
      3F9E2DA8#)
   (x #4EB9993416934FAE476E4655B5A520373F1321CE#)
   )
```

The fingerprint is calculated (leaving out the 0x0000 Key Type for DSA) by hashing:

```
    SHA-1( p(MPI) | q(MPI) | g(MPI) | y(MPI) )
```

which yields (in hexadecimal):

```
    35b3c7c02cf9e74bd53f33a0bb815ccd39e60a8d
```

Resulting in the following OTRFP record:

```
    nb2wo2a=._otrfp.example.com. IN OTRFP (
        3 0 1 35b3c7c02cf9e74bd53f33a0bb815ccd39e60a8d )
```

7. Acknowledgements

This document is based on RFC-4255 and draft-ietf-dane-smime whose authors are Paul Hoffman, Jacob Schlyter and W. Griffin. The MPI section has been taken verbatim from RFC-4880.

8. References

8.1. Normative References

[OTRSPEC] , "Off-the-Record Messaging Protocol version 3", September 2012, <http://www.cypherpunks.ca/otr/Protocol-v3-4.0.0.html>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, January 2006.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[XMPP] , "XMPP Standards Foundation RFCs", February 2003, <http://xmpp.org/xmpp-protocols/rfcs/>.

8.2. Informative References

[RFC2812] Kalt, C., "Internet Relay Chat: Client Protocol", RFC 2812, April 2000.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

Author's Address

Paul Wouters
Red Hat

Email: pwouters@redhat.com