Network Working Group   J. Levine
Internet-Draft   Taughannock Networks
Updates: 6376 (if approved)   July 27, 2017
Intended status: Standards Track
Expires: January 28, 2018

New cryptographic signature methods for DKIM
draft-ietf-dcrup-dkim-crypto-04

Abstract

DKIM was designed to allow new cryptographic algorithms to be added. This document adds a new signing algorithm and a new way to represent signature validation keys, and deprecates an obsolete signing algorithm.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Discussion Venue: Discussion about this draft is directed to the dcrup@ietf.org [1] mailing list.

DKIM [RFC6376] signs e-mail messages, by creating hashes of the message headers and content and signing the header hash with a digital signature. Message recipients fetch the signature verification key from the DNS where it is stored in a TXT record. The defining documents specify a single signing algorithm, RSA [RFC3447], and recommends key sizes of 1024 to 2048 bits. While 1024 bit signatures are common, stronger signatures are not. Widely used DNS configuration software places a practical limit on key sizes, because the software only handles a single 256 octet string in a TXT record, and RSA keys longer than 1156 bits don't fit in 256 octets.

This document adds a new stronger signing algorithm, Edwards-Curve Digital Signature Algorithm using the Curve25519 curve (ed25519), which has much shorter keys than RSA for similar levels of security. It also adds a new key representation for RSA keys, with the key itself in the signature and a key fingerprint that fits in a a 256 octet string in the DNS regardless of the key length.

2. Conventions Used in This Document

The capitalized 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 [RFC2119].

Syntax descriptions use Augmented BNF (ABNF) [RFC5234]. The ABNF tokens sig-a-tag-k, key-k-tag-type, and base64string are imported from [RFC6376].

3. Public key fingerprints

Rather than using a public key stored in the DNS, an RSA signature can include the corresponding public key, with a verification fingerprint in the DNS. For an RSA signature with a key fingerprint, the Signing Algorithm is rsafp-sha256. The DNS record contains a SHA-256 hash of the public key, stored in base64 in the p= tag. The key type tag MUST be present and contains k=rsafp.

Note: since Ed25519 keys are 256 bits long, a SHA-256 hash of a key is the same size as the key itself, so there would be no benefit to storing ed25519 key fingerprints in the key record rather than the keys themselves.

Section 5.5 of [RFC6376], on computing the message hash and signature, is modified as follows: When creating a signature with a signing algorithm that uses a key fingerprint, the signer includes the public key in the signature as a base64 encoded string with a k= tag. The key in the tag is the same one that would be published in a non-fingerprint key record.

Section 3.7 of [RFC6376], on computing the message hashes, is not modified. Since the key in the k= tag is known in advance, it included in the signature in the same manner as all of the other signature fields other than b=.

Section 6.1.3 of [RFC6376], to compute the verification, is modified as follows: In item 4, if the signing algorithm uses a key fingerprint, extract the verification key from the k= tag. If there is no such tag, the signature does not validate. Extract the key hash from the p= tag of the key record. If there is no such tag or the tag is empty, the signature does not validate. Compute the SHA-256 hash of the verification key, and compare it to the value of the key hash. If they are not the same, the signature does not validate. Otherwise proceed to verify the signature using the validation key and the algorithm described in the "a=" tag.

4. Ed25519-SHA256 Signing Algorithm

The ed25519-sha256 signing algorithm computes a message hash as defined in section 3 of [RFC6376], and signs it with the Pure variant of Ed25519, as defined in in RFC 8032 section 5.1 [RFC8032]. The signing algorithm is PureEdDSA, since the input to the signing algorithm has already been hashed. The DNS record for the verification public key MUST have a "k=ed25519" tag to indicate that the key is an Ed25519 rather than RSA key.

The signature MAY contain a base64 copy of the public key in the p= tag, to enable checks for a cryptograhphic attack described in the Security section below.

5. Signature and key syntax

The syntax of DKIM signatures and DKIM keys are updated as follows.

5.1. Signature syntax

The syntax of DKIM algorithm tags in section 3.5 of [RFC6376] is updated by adding this rule to the existing rule for sig-a-tag-k:

```
ABNF:

sig-a-tag-k =/ "rsafp" / "ed25519"
```

The following tag is added to the list of tags on the DKIM-Signature header field in section 3.5 of [RFC6376].

k= The public key (base64; REQUIRED). White space is ignored in this value and MUST be ignored when reassembling the original key.

```
ABNF:

sig-k-tag       = %x6b [FWS] "=" [FWS] sig-k-tag-data
sig-k-tag-data  = base64string
```

5.2. Key syntax

The syntax of DKIM key tags in section 3.6.1 of [RFC6376] is updated by adding this rule to the existing rule for key-k-tag-type:

```
ABNF:

key-k-tag-type  =/ "rsafp" / "ed25519"
```

6. Key and algorithm choice and strength

Section 3.3 of [RFC6376] describes DKIM's hash and signature algorithms. It is updated as follows:

Signers SHOULD implement and verifiers MUST implement the rsafp-sha256 and ed25519-sha256 algorithms.

Signers that use rsa-sha256 or rsafp-sha256 signatures MUST use keys at least 1024 bits long and SHOULD use keys 2048 bits long. Verifiers SHOULD NOT accept rsa-sha256 or rsafp-sha256 signatures with keys less than 1024 bits long.

7. Transition Considerations

For backward compatibility, signers MAY add multiple signatures that use old and new signing algorithms or key representations. Since there can only be a single key record in the DNS for each selector, the signatures will have to use different selectors, although they can use the same d= and i= identifiers.

8. Security Considerations

Ed25519 and key fingerprints are widely used cryptographic techniques, so the security of DKIM signatures using new signing algorithms should be at least as good as those using old algorithms. Since key fingerprints make it possible to publish verification records for RSA keys of any length, rsafp signatures SHOULD use key lengths of 1536 or 2048 bits.

8.1. Duplicate Signature Key Selection

The rsafp signature scheme describes a method where the public key is hashed. The primary motivation for this design is allowing for a smaller key representation of larger public keys. Hashing has a secondary effect: the public key is included in messages and is signed. Including and signing the public key renders duplicate signature key selection attacks [Koblitz13] infeasible. An attacker cannot claim a message by constructing a key that would be valid for a signed message because the public key is covered by the signature.

There is currently no known way that an attacker might use a duplicate signature key selection attack to their advantage and so defending against the attack is not mandated by this specification. In the event that a potential attack becomes known, a signer could include the public key in messages using the `p=` parameter. If the `p=` parameter is present, a verifier MUST ensure that the parameter contains the public key that it uses to verify the message signature.

9. IANA Considerations

IANA is requested to update registries as follows.

9.1. DKIM Signature Tag Registry

The following value is added to the DKIM Signature Tag Registry

TYPE	REFERENCE	STATUS
k	(this document)	active

Table 1: DKIM Signature Tag Registry Added Value

9.2. DKIM Key Type registry

The following values are added to the DKIM Key Type Registry

TYPE	REFERENCE	STATUS
rsafp	[RFC3447]	active
ed25519	[RFC8032]	active

Table 2: DKIM Key Type Registry Added Values

10. References

10.1. Normative References

[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>.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, DOI 10.17487/RFC3447, February 2003, <http://www.rfc-editor.org/info/rfc3447>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <http://www.rfc-editor.org/info/rfc6376>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <http://www.rfc-editor.org/info/rfc8032>.

10.2. Informative References

[Koblitz13] Koblitz, N. and A. Menezes, "Another look at security definitions", DOI 10.3934/amc.2013.7.1, 2013, <http://aimsciences.org/journals/displayArticlesnew.jsp?paperID=8249>.

Advances in Mathematics of Communications, Vol 7, Num 1, pp. 1-38.

10.3. URIs

[1] mailto:dcrup@ietf.org

Appendix A. Change log

03 to 04: Change eddsa to ed25519. Add Martin's key regeneration issue. Remove hashed ed25519 keys. Fix typos and clarify text. Move syntax updates to separate section. Take out SHA-1 stuff.

01 to 02: Clarify EdDSA algorithm is ed25519 with Pure version of the signing. Make references to tags and fields consistent.

Author's Address

John Levine
Taughannock Networks
PO Box 727
Trumansburg, NY 14886

Phone: +1 831 480 2300
Email: standards@taugh.com