Workgroup: openpgp
Internet-Draft: draft-dkg-openpgp-1pa3pc-00
Published: 19 August 2023
Intended Status: Informational
Expires: 20 February 2024
Authors: D. K. Gillmor
ACLU

First-Party Attested Third-Party Certifications in OpenPGP

Abstract

An OpenPGP certificate can grow in size without bound when third-party certifications are included. This document describes a way for the owner of the certificate to explicitly approve of specific third-party certifications, so that relying parties can safely prune the certificate of any unapproved certifications.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://dkg.gitlab.io/openpgp-1pa3pc/. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dkg-openpgp-1pa3pc/.

Discussion of this document takes place on the OpenPGP Working Group mailing list (mailto:openpgp@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/openpgp/. Subscribe at https://www.ietf.org/mailman/listinfo/openpgp/.

Source for this draft and an issue tracker can be found at https://gitlab.com/dkg/openpgp-1pa3pc.

Status of This Memo

This Internet-Draft will expire on 20 February 2024.

Copyright Notice

Table of Contents

1. Introduction

In some cases, it is useful to have a third-party certification over an identity in an OpenPGP certificate. However, if an OpenPGP certificate simply merges in all third-party certifications, the certificate can grow in size to the point where it is impossible to use or transfer. See, for example, the discussion about "certificate flooding" in Section 2.1 of [I-D.dkg-openpgp-abuse-resistant-keystore].

If the owner of an OpenPGP certificate (the "keyholder") wants their own certificate to be usable by others, they can explicitly indicate which third-party certifications they approve of, and implicitly decline the rest.

1.1. Terminology

The term "OpenPGP Certificate" is used in this document interchangeably with "OpenPGP Transferable Public Key", as defined in Section 10.1 of [I-D.ietf-openpgp-crypto-refresh].

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Wire Format

This specification defines a new signature type, a new signature subpacket type, and extends the structure of an OpenPGP certificate.

2.1. Attestation Key Signature

This document defines a new key signature type used only in OpenPGP certificates known as an Attestation Key Signature. The Signature type ID is 0x16.

This signature is issued by the primary key over itself and its User ID (or User Attribute). It **MUST** contain exactly three subpackets in its hashed subpackets:

- a "Signature Creation Time" subpacket (Section 5.2.3.11 of [I-D.ietf-openpgp-crypto-refresh])
- an Issuer Fingerprint subpacket (see Section 5.2.3.35 of [I-D.ietf-openpgp-crypto-refresh])
- an "Attested Certifications" subpacket (see Section 2.2)

This type of key signature does not replace or override any standard certification (0x10-0x13).

Only the most recent self-signed Attestation Key Signature is valid for any given <key,userid> pair. If more than one self-signed Attestation Key Signature is present with the same Signature Creation Time, the set of attestations should be treated as the union of all "Attested Certifications" subpackets from all such signatures with the same timestamp.

2.1.1. Computing an Attestation Key Signature

An Attestation Key Signature is computed over a hash of data in the same way as a Certification Signature. That is, the following items are concatenated into the hash function before signing:

- The salt (or nothing at all, if the signature version is less than 6)
- The serialized Primary Key
- The serialized User ID
- The trailer, which includes the signature data including the hashed subpackets

2.2. Attested Certifications Subpacket

This document defines a new signature subpacket named Attested Certifications. Its contents are N octets of certification digests (see more below).

This subpacket **MUST** only appear as a hashed subpacket of an self-signed Attestation Key Signature (see Section 2.1). It has no meaning in any other signature type. It is used by the primary key to attest to a set of third-party certifications over the associated User ID or User Attribute. This enables the holder of an OpenPGP primary key to mark specific third-party certifications as re-distributable with the rest of the Transferable Public Key (see the "No-modify" flag in Section 5.2.3.25 of [I-D.ietf-openpgp-crypto-refresh]). Implementations **MUST** include exactly one Attested Certification subpacket in any generated Attestation Key Signature.

The contents of the subpacket consists of a series of digests using the same hash algorithm used by the signature itself. Each digest is made over one third-party signature (any Certification, i.e., signature types 0x10-0x13) that covers the same Primary Key and User ID (or User Attribute). For example, an Attestation Key Signature made by key X over User ID U using hash algorithm SHA256 might

contain an Attested Certifications subpacket of 192 octets (6*32 octets) covering six third-party certification Signatures over <X,U>. They **SHOULD** be ordered by binary hash value from low to high (e.g., a hash with hexadecimal value 037a… precedes a hash with value 0392…, etc). The length of this subpacket **MUST** be an integer multiple of the length of the hash algorithm used for the enclosing Attestation Key Signature.

The listed digests **MUST** be calculated over the third-party certification's Signature packet as described in Section 5.2.4 of [I-D.ietf-openpgp-crypto-refresh], but without a trailer: the hash data starts with the octet 0x88, followed by the four-octet length of the Signature, and then the body of the Signature packet. (Note that this is an Legacy Format packet header for a Signature packet with the length-of-length field set to zero.) The unhashed subpacket data of the Signature packet being hashed is not included in the hash, and the unhashed subpacket data length value is set to zero.

If an implementation encounters more than one such subpacket in an Attestation Key Signature, it **MUST** treat it as a single Attested Certifications subpacket containing the union of all hashes.

The Attested Certifications subpacket in the most recent self-signed Attestation Key Signature over a given User ID supersedes all Attested Certifications subpackets from any previous Attestation Key Signature. However, note that if more than one Attestation Key Signature packets have the same (most recent) Signature Creation Time subpacket, implementations **MUST** consider the union of the attestations of all Attestation Key Signatures. This allows the keyholder to attest to more third-party certifications than could fit in a single Attestation Key Signature.

Note that Certification Revocation Signatures are not relevant for Attestation Key Signatures. To rescind all attestations, the primary key holder needs only to publish a more recent Attestation Key Signature with an empty Attested Certifications subpacket.

2.3. Placement in OpenPGP Certificate

The Attestation Key Signature appears in an OpenPGP certificate after a User ID or User Attribute packet, mixed in with the certifications that cover that User ID or User Attribute packet.

FIXME: test that these do not break existing implementations by causing them to reject a certificate that they otherwise would have accepted. If they do, then we might consider placing this signature in an unhashed Embedded Signature subpacket in the User ID's self-sig.

3. Semantics

The inclusion of a digest in an Attested Certifications subpacket in a valid, most-recent self-signed Attestation Key signature which matches a specific third-party certification is an indication that the keyholder approves of the third-party certification.

There is no need to attest to self-signed certifications. Since they are already made by the primary key, self-signed certifications are implicitly approved.

A verifier might observe a attested digest that does not correspond to any Certification that the verifier is aware of. This is normal, because not everyone is guaranteed to have the exact same set of third-party certifications for any given OpenPGP certificate. In such cases, the verifier should ignore the non-matching digest, but **MUST NOT** ignore other digests in the list of Attested Certifications.

4. Reasonable Workflows

This section describes some possible steps for generating and using Attested Certifications.

4.1. Third-party Certification and Attestation Workflow

Alice has a new OpenPGP certificate with primary key K, and wants to publish Bob's certification over her User ID in that certificate.

Alice sends Bob her certificate, asking for his certification. Bob performs his normal verification that the User ID and K do indeed belong to Alice, and then creates a certification over her User ID, adding it to the certificate.

Bob then sends the augmented certificate back to Alice. Alice reviews the added certification, and decides that she likes it.

She chooses a strong hash algorithm H and uses it to compute the digest of Bob's certification. She places that digest into an Attested Certifications subpacket S. She also creates a Signature Creation Time subpacket C containing the current timestamp, and an Issuer Fingerprint subpacket F containing the fingerprint of K.

Alice places subpackets F, C, and S into an Attestation Key Signature packet, and signs it with K using hash algorithm H.

4.2. Keyholder Update Workflow

If a keyholder Alice has already attested to third-party certifications from Bob and Carol and she wants to add an

attestation to a certification from David, she should issue a new Attestation Key Signature (with a more recent Signature Creation timestamp) that contains an Attested Certifications subpacket covering all three third-party certifications.

If she later decides that she does not want Carol's certification to be redistributed with her certificate, she can issue a new Attestation Key Signature (again, with a more recent Signature Creation timestamp) that contains an Attested Certifications subpacket covering only the certifications from Bob and David.

4.3. Distributor Workflow

If an abuse-resistant keystore (e.g., an OpenPGP keyserver) receives an OpenPGP certificate for redistribution, it **SHOULD** strip away all unattested third-party certifications before redistributing the certificate.

If such a keystore receives an updated copy of the certificate which includes a newer Attestation Key Signature, it should merge the certificate update with its existing copy of the certificate, and re-apply the new list of attested digests by stripping away all certifications which do not match the new list.

5. Security Considerations

This document is intended to make an OpenPGP certificate more manageable by the keyholder.

A flooded certificate is difficult or impossible to redistribute, which means that peers of the keyholder cannot easily fetch the certificate, resulting in inability to encrypt messages to or verify signatures from that certificate. An unredistributable certificate can also make it difficult or impossible to transmit revocation, expiration, key rotation, or preference changes associated with the certificate, which interferes with certificate maintenance necessary to securely use OpenPGP.

The mechanisms described in this document defend against certificate flooding attacks by enabling certificate redistributors (e.g., keyserver networks or other "keystores") to limit the contents of a certificate to only those elements which the keyholder explicitly approves of and wants included in the certificate.

6. IANA Considerations

IANA is asked to register multiple objects in the OpenPGP protocol group.

6.1. Signature Types: Add Attestation Key Signature

The Signature Types registry should add a row with signature type 0x16, Name "Attestation Key Signature", and Reference pointing to Section 2.1 in this document.

6.2. Signature Subpacket Type: Attested Certifications

The Signature Subpacket Types registry row with Type 37 should be update with Description "Attested Certifications", and Reference pointing to Section 2.2 in this document.

7. References

7.1. Normative References

[I-D.ietf-openpgp-crypto-refresh] Wouters, P., Huigens, D., Winter, J., and N. Yutaka, "OpenPGP", Work in Progress, Internet-Draft, draft-ietf-openpgp-crypto-refresh-10, 21 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-openpgp-crypto-refresh-10>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7.2. Informative References

[I-D.dkg-openpgp-abuse-resistant-keystore] Gillmor, D. K., "Abuse-Resistant OpenPGP Keystores", Work in Progress, Internet-Draft, draft-dkg-openpgp-abuse-resistant-keystore-06, 18 August 2023, <https://datatracker.ietf.org/api/v1/doc/document/draft-dkg-openpgp-abuse-resistant-keystore/>.

Appendix A. Augmenting SOP For 1PA3PC

FIXME: Can all of the plausible workflows described in this document be done with the Stateless OpenPGP Interface? Definitely not right now. What is missing?

Appendix B. Test Vectors

FIXME: This document should include a certificate with third-party certifications, some of which are approved, and others of which are

not approved. It should also show the same certificate, but pruned to remove all non-approved third-party certifications.

Appendix C. Existing Implementations

RFC Editor Note: Please delete this section before publication.

FIXME: enumerate existing implementations.

Appendix D. Acknowledgements

Demi Marie Obenour, Heiko Stamer, Jan Zerebecki, Justus Winter, Neal Walfield, Vincent Breitmoser, and others all contributed to specifying and defining this mechanism.

Appendix E. Substantive changes to this document

RFC Editor Note: Please delete this section before publication.

E.1. Substantive Changes from MR !60 to draft-dkg-openpgp-1pa3pc-00

* https://gitlab.com/openpgp-wg/rfc4880bis/-/merge_requests/60 describes the earlier draft of this proposal.

* This draft transcribes most of that MR, updating references and including explicit IANA considerations.

Author's Address

Daniel Kahn Gillmor
ACLU

Email: dkg@fifthhorseman.net