Workgroup: openpgp
Internet-Draft:
draft-gallagher-openpgp-literal-metadata-00
Updates: 4880 (if approved)
Published: 2 January 2024
Intended Status: Informational
Expires: 5 July 2024
Authors: A. Gallagher, Ed.
PGPKeys.EU

OpenPGP Literal Data Metadata Integrity

Abstract

This document specifies a method for ensuring the integrity of file metadata when signed using OpenPGP.

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://andrewgdotcom.gitlab.io/draft-gallagher-openpgp-literal-metadata. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-gallagher-openpgp-literal-metadata/.

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/andrewgdotcom/draft-gallagher-openpgp-literal-metadata.

Status of This Memo

Table of Contents

1. Introduction

Due to a design oversight in an early version of PGP, the literal data metadata (file type, file name, and file timestamp) were not covered by any integrity-protection mechanisms. That omission has persisted through subsequent OpenPGP specifications, up to and including [RFC4880]. This document introduces the missing integrity check by adopting and extending the "Literal Data Meta Hash" subpacket from [LIBREPGP], section 5.2.3.33.

2. Conventions and Definitions

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.

3. Literal Data Meta subpacket

This subpacket **MAY** be used to protect the metadata of a Literal Data Packet. It is only useful when located in the hashed-subpackets area of a v4 (or later) signature over a Literal Data Packet, and so it **SHOULD NOT** appear elsewhere. The packet **SHOULD** be marked as critical.

The metadata is always of the same form as it appears in the Literal Data Packet (type 11, [RFC4880] section 5.9), i.e.:

- A one-octet field that describes how the data is formatted.
- File name as a string (one-octet length, followed by a file name).
- A four-octet number that indicates a date associated with the literal data.

The first octet of the subpacket contents indicates the encoding of the subpacket. A value of 0 indicates Hashed encoding, and a value of 1 indicates Verbatim encoding. The remaining octets of the subpacket are encoding-dependent.

3.1. Literal Data Meta (Hashed)

The remainder of the packet contains a 32 octet fixed-length hash value. The hash is computed over the metadata as specified above, using SHA-256 [FIPS180]. When an implementation is validating a signature containing a Literal Data Meta (Hashed) subpacket in its hashed-subpackets area, it **MUST** re-create the hash from the metadata section of the Literal Data packet that is signed over. If the calculated hash value does not match the one in the subpacket, the signature **MUST** be deemed as invalid.

3.2. Literal Data Meta (Verbatim)

The remainder of the packet contains a verbatim copy of the metadata as specified above. When an implementation has successfully validated a signature containing a Literal Data Meta (Verbatim) subpacket in its hashed-subpackets area, the metadata section of the Literal Data Packet that is signed over **MUST** be replaced with the copy from the Literal Data Meta subpacket.

4. Security Considerations

A signature containing a Literal Data Meta (Verbatim) subpacket can be round-tripped via detached-signature format without loss of integrity. A signature containing a Literal Data Meta (Hashed) subpacket does not have this property, because the metadata cannot

be regenerated from the hash and is therefore lost on conversion to a detached signature.

It is therefore **RECOMMENDED** that implementations generate Literal Data Hash subpackets using the Verbatim encoding.

5. IANA Considerations

This document requests that the following entry be added to the OpenPGP Signature Subpacket registry:

Type	Name	Specification
40	Literal Data Meta	This document

Table 1: Signature Subpacket Registry

6. References

6.1. Normative References

[FIPS180] National Institute of Standards and Technology, U.S. Department of Commerce, "Secure Hash Standard (SHS), FIPS 180-4", August 2015, <http://dx.doi.org/10.6028/NIST.FIPS.180-4>.

[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/rfc/rfc2119>.

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

[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/rfc/rfc8174>.

6.2. Informative References

[LIBREPGP] Koch, W. and R. H. Tse, "LibrePGP Message Format", 2023, <https://datatracker.ietf.org/doc/html/draft-koch-librepgp>.

Appendix A. Acknowledgments

The author would like to thank Werner Koch for his earlier work on the Literal Data Meta Hash subpacket.

Author's Address

Andrew Gallagher (editor)
PGPKeys.EU

Email: andrewg@andrewg.com