Network Working Group
Internet-Draft
Updates: 9580 (if approved)
Intended status: Standards Track
Expires: 14 August 2026

D. Huigens, Ed.
Proton AG

10 February 2026

Persistent Symmetric Keys in OpenPGP
draft-ietf-openpgp-persistent-symmetric-keys-03

Abstract

This document defines a new packet and algorithm for the OpenPGP standard (RFC 9580) to support persistent symmetric keys, for message encryption using authenticated encryption with additional data (AEAD) and for message authentication using AEAD authentication tags. This enables the use of symmetric cryptography for data storage (and other contexts that do not require asymmetric cryptography), for improved performance, smaller keys, and improved resistance to quantum computing.

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://twisstle.gitlab.io/openpgp-persistent-symmetric-keys/. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-openpgp-persistent-symmetric-keys/.

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/twisstle/openpgp-persistent-symmetric-keys.

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 https://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 14 August 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

The OpenPGP standard [RFC9580] has supported symmetric encryption for data packets using session keys since its inception, as well as symmetric encryption using password-derived keys. This document extends the use of symmetric cryptography by adding support for persistent symmetric keys which can be stored in a transferable secret key, and used to symmetrically encrypt session keys, for long-term storage and archival of messages. This document uses authenticated encryption with associated data (AEAD) as defined by [RFC9580].

The OpenPGP standard also supports the use of digital signatures for authentication and integrity but no similar symmetric mechanism exists in the standard. This document introduces the use of AEAD authentication tags as a symmetric counterpart to digital signatures, for long-term storage and archival of attestations of authenticity and certification.

2. Conventions Used in This Document

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 [RFC2119]. Any implementation that adheres to the format and methods specified in this document is called a compliant application. Compliant applications are a subset of the broader set of OpenPGP applications described in [RFC9580]. Any [RFC2119] keyword within this document applies to compliant applications only.

3. Motivation

When compared to asymmetric cryptography, symmetric cryptography can provide improved performance and equivalent security with smaller keys. In contexts that do not require asymmetric cryptography, such as secure data storage where the same user encrypts and decrypts data, symmetric cryptography can be used to take advantage of these benefits.

Additionally, asymmetric algorithms included in OpenPGP are vulnerable to attacks that might become possible on quantum computers [Shor]. Symmetric cryptography is also affected by quantum computing but to a lesser extent, which can be countered by using larger keys [Grover]. While the standardization of quantum-secure asymmetric

cryptography in OpenPGP is ongoing [PQCinOpenPGP], and will be required to secure communications, there is a large body of existing messages encrypted with classical algorithms. Once persistent symmetric keys are available, these messages can be protected against future compromises efficiently by symmetrically re-encrypting the session key, and storing the message symmetrically encrypted for long-term storage and archival.

4. Creating and Reusing Packets

For storing persistent symmetric keys, we introduce a new packet (see Section 5), as handling of persistent symmetric key material requires some care. For example, when extracting a Transferable Public Key from a Transferable Secret Key, persistent symmetric keys must be ignored.

For storing session keys encrypted with persistent symmetric keys, while a Symmetric-Key Encrypted Session Key packet exists, its semantics don't match our requirements, as it's intended to encrypt the session key with a user-provided password, and doesn't offer a way to store a reference to a persistent key. Therefore, we reuse the Public-Key Encrypted Session Key packet instead, which does offer the desired semantics.

Similarly, we reuse the Signature packet for "symmetric signatures".

For these use cases, no new packet is required as the handling of these packets requires no special care.

To indicate the type of keys in the new and existing packets, a special persistent symmetric algorithm ID value 0 is registered, which can be used in the place of a public-key algorithm ID.

5. Persistent Symmetric Key Packet

This document defines a new OpenPGP packet, extending table 3 of [RFC9580].

ID	Critical	Packet Type Description	Shorthand
40	No	Persistent Symmetric Key Packet	PSK

Table 1: Persistent Symmetric Key Packet registration

The Persistent Symmetric Key Packet (Type ID 40) has identical fields to the Secret Key Packet (Type ID 5). However, only version 6 of the packet is defined. Earlier versions of the Secret Key Packet format MUST NOT be used with the Persistent Symmetric Key Packet.

The Persistent Symmetric Key Packet MUST NOT be used with asymmetric algorithms, i.e. any of the public key algorithms defined in table 18 of [RFC9580]. It may only be used with the persistent symmetric algorithm defined below, with special algorithm ID value 0.

When storing encrypted symmetric key material in a Persistent Symmetric Key Packet, AEAD encryption (S2K usage octet 253, see section 3.7.2.1 of [RFC9580]) MUST be used, to ensure that the secret key material is bound to the fingerprint. Implementations MUST NOT decrypt symmetric key material in a Persistent Symmetric Key Packet that was encrypted using a different method.

6. Extended Transferable Secret Key Grammar

The Transferable Secret Key grammar defined in section 10.2 of [RFC9580] is extended to allow including Persistent Symmetric Key Packets. Persistent Symmetric Keys may be included together with asymmetric keys in a single sequence of Transferable Secret Keys. However, Persistent Symmetric Keys do not accept subcomponents, such as subkeys, User IDs, or direct-key signatures.

7. Persistent Symmetric Key Algorithm

This document defines one new algorithm for use with OpenPGP, updating table 18 of [RFC9580].

ID	Algorithm	Public Key Format	Secret Key Format	Signature Format	PKESK Format
0	AEAD	sym. algo, fingerprint, seed [Section 7.1]	key material	AEAD algo, salt, authentication tag [Section 7.3]	AEAD algo, salt, ciphertext [Section 7.2]

Table 2: Persistent Symmetric Key Algorithm registration

This algorithm ID can be used to store symmetric key material in a Persistent Symmetric Key Packet (see Section 5). It can also be used to store a session key encrypted using AEAD in a PKESK packet (see

section 5.1 of [RFC9580]). It can also be used to store an AEAD authentication tag in a Signature packet (see section 5.2 of [RFC9580]).

Implementations MUST NOT use the symmetric algorithm ID in Public-Key Packets, Public-Subkey Packets, Secret-Key Packets or Secret-Subkey Packets.

7.1. Algorithm-Specific Fields for Persistent Symmetric Keys

The public key material consists of this series of values:

* A 1-octet symmetric algorithm ID (see section 9.3 of [RFC9580]).

* A 32-octet random seed to randomize the key fingerprint.

The secret key material consists of this single value:

* Symmetric key material of the appropriate length for the given symmetric algorithm.

7.2. Algorithm-Specific Fields for Persistent Symmetric Encryption

The encrypted session key consists of this series of values:

* A 1-octet AEAD algorithm ID (see section 9.6 of [RFC9580]).

* 32 octets of salt. The salt is used to derive the key-encryption key and MUST be securely generated (see section 13.10 of [RFC9580]).

* The ciphertext and authentication tag resulting from a symmetric key encryption performed using the symmetric-key cipher of the key and the indicated AEAD mode, with the key-encryption key and IV computed as described in Section 7.4, and as additional data the empty string. The following values are concatenated and then encrypted:

  - The 1-octet algorithm identifier, if it was passed (in the case of a v3 PKESK packet).

  - The session key described in section 5.1 of [RFC9580].

7.3. Algorithm-Specific Fields for Persistent Symmetric Signatures

The signature consists of this series of values:

* A 1-octet AEAD algorithm ID (see section 9.6 of [RFC9580]).

* 32 octets of salt. The salt is used to derive the message authentication key and MUST be securely generated (see section 13.10 of [RFC9580]).

* The authentication tag resulting from encrypting the empty string using the symmetric-key cipher of the key and the indicated AEAD mode, with the message authentication key and IV computed as described in Section 7.4, and as additional data the hash digest described in section 5.2.4 of [RFC9580].

Although not required by AEAD algorithms, to maintain consistency with existing signature algorithms, AEAD authentication tags are produced from appropriately hashed data, as per section 5.2.4 of [RFC9580].

7.4. Key and IV derivation

When encrypting a session key or authenticating a message using a persistent symmetric key, the persistent key material and the salt are used to derive an M-bit key and N bits used as initialization vector, where M is the key size of the symmetric algorithm and N is the nonce size of the AEAD algorithm.

M + N bits are derived using HKDF (see [RFC5869]). The left-most M bits are used as symmetric algorithm key, the remaining N bits are used as initialization vector. HKDF is used with SHA512 ([RFC6234]) as hash algorithm, the persistent key material as Initial Keying Material (IKM), the salt as salt, and the Packet Type ID in OpenPGP format encoding (bits 7 and 6 set, bits 5-0 carry the packet type ID), packet version number, cipher algorithm ID and AEAD algorithm ID as info parameter.

For example, for a version 6 PKESK packet, when using AES-128 in OCB mode, the info parameter would be the octets 0xC1, 0x06, 0x07, 0x02. For a version 6 Signature packet, when using AES-256 in GCM mode, the info parameter would be the octets 0xC2, 0x06, 0x09, 0x03.

8. Security Considerations

Security considerations are discussed throughout the document where appropriate.

9. IANA Considerations

9.1. Updates to Packet Types

IANA is requested to update the "OpenPGP Packet Types" registry with the entry in Table 1.

9.2. Updates to Public Key Algorithms

IANA is requested to update the "OpenPGP Public Key Algorithms" registry with the entry in Table 2.

10. Acknowledgements

An initial version of this draft was written by Dan Ristea (Proton AG), with guidance from Dr Philipp Jovanovic (University College London) and the editor.

Thanks to feedback and suggestions from Andrew Gallagher, Heiko Sch辰fer, Justus Winter, Falko Strenzke, Daniel Kahn Gillmor, and Lara Bruseghini, in no particular order.

11. References

11.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC9580] Wouters, P., Ed., Huigens, D., Winter, J., and Y. Niibe, "OpenPGP", RFC 9580, DOI 10.17487/RFC9580, July 2024, <https://www.rfc-editor.org/info/rfc9580>.

11.2. Informative References

[Grover] Grover, L., "Quantum mechanics helps in searching for a needle in a haystack", 1997, <https://arxiv.org/abs/quant-ph/9706033>.

[PQCinOpenPGP]
Kousidis, S., Strenzke, F., and A. Wussler, "Post-Quantum Cryptography in OpenPGP", October 2023, <https://datatracker.ietf.org/doc/html/draft-wussler-openpgp-pqc-03>.

[Shor] Shor, P., "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", October 1997, <http://dx.doi.org/10.1137/S0097539795293172>.

Appendix A. Test Vectors

To help implementing this specification a set of non-normative examples follow here.

A.1. Transferable Secret Key

Here is a Transferable Secret Key consisting of:

* A v6 Persistent Symmetric Key packet

```
-----BEGIN PGP PRIVATE KEY BLOCK-----

6DwGaXu4NQAAAAAhB778G6ol+0zgePhnCmEi6PdPZm/BNNe8ao/8MxvEOPno
AE+GgapnjN6EZboLhSVqQL0=
-----END PGP PRIVATE KEY BLOCK-----
```

The key has the fingerprint eeea1e834ed8aacf3d938a5ccc6a177fcb7775c290eb02c3c17e5fdaa559b2a5.

The raw key material is 4f8681aa678cde8465ba0b85256a40bd (for AES-128).

A.2. V6 Encrypted Message

Here is the message "Hello World" encrypted using the secret key in Appendix A.1, consisting of:

* A v6 PKESK packet

* A v2 SEIPD packet

```
-----BEGIN PGP MESSAGE-----

wXUGIQbu6h6DTtiqzz2TilzMahd/y3d1wpDrAsPBfl/apVmypQACWjxc6uw6
l22qqqSVwJBxSZ03nXQYghPddk5PQJGHQ8L2bADc0Me4GKcpz1SEfGaRxAqt
fkmO2J4WRfPr3ciaacmwbL2EE4TWbUC3nKnyaeHSVwIJAwzL9pRdO5Sazs1T
1f6fl9MfUD/3NqbADxVAiGMbw7e3QbKrMsX0ulYJRsi61DQVummllYExiVnt
K7uBUriICKUzqo+8Q2V8HpMeyV+audJkp62Yuw==
-----END PGP MESSAGE-----
```

The session key is encrypted using AES-128 in OCB mode. The HKDF info is c1060702. The key-encryption key is a64e6ac3aa262e42da23d5a14470857b. The IV is 2d9c726a1b01f60f3b05b89cb887f9. The additional data is the empty string.

The encrypted session key is 25e5884dc13bcf40d5ab41b3bc23306ebb1907b4d586b2d678e21c496a8bf35a. The message is encrypted using AES-256 in GCM mode.

A.3. V3 Encrypted Message

Here is the message "Hello World" encrypted using the secret key in Appendix A.1, consisting of:

* A v3 PKESK packet

* A v1 SEIPD packet

```
-----BEGIN PGP MESSAGE-----

wVwD7uoeg07Yqs8AAuvkkq5zcghyUHMUZa6aZS90Ro/C+7vJFoRjkKCUKRCa
DcKFbmZ3oFj3lq28Ibe8hkTfqr8SLDTcl0P7TnkufzeHBvH6AwPYNCEhvPKS
yyoL/tI8AThJZhgKkUZDFTKhU8CY4r7dCpbAUGz4wjVMseFDKZOAc+ywKt4s
nrX5PNU9VtHTO1ceJmJJq2DKrH+Y
=+t4C
-----END PGP MESSAGE-----
```

The session key is encrypted using AES-128. The HKDF info is c1030702. The key-encryption key is 0c233981b8481ff68d253e78e9d7fc1b. The IV is 6186c39f7673f2d540f193e05ecab0. The additional data is the empty string.

The encrypted session key is 5f43304125ba1728aab843c0a6d4ce75bd8eb3961a2025c3131e58223ecafee2. The message is encrypted using AES-256.

A.4. Detached Signature

Here is a detached signature over the message "Hello World" signed using the secret key in Appendix A.1, consisting of:

* A v6 Signature packet

```
-----BEGIN PGP SIGNATURE-----

wnkGAAAIAAAAKQWCaXu9QCKhBu7qHoNO2KrPPZOKXMxqF3/Ld3XCkOsCw8F+
X9qlWbKlAAAAADI0EHT4wSYfRzYXBKpTBs1pmc0Cb+a1BVcMwGMjrKgN1hS4
4R9cgg2C385fWUj4w2KGk7dVXpMKX1Yj+WvErkO5slZd
-----END PGP SIGNATURE-----
```

The message is authenticated using AES-128 in OCB mode. The HKDF info is c2060702. The authentication key is 31988c7b45116e19b24ff29a1b93d4c2. The IV is 682aa0b80b031fe8882cecf1b6dca6. The additional data is 32342abfe99d053a5ea192b9192fb88210f6c479bc6a7be6e653e34147593bac.

Author’s Address

Daniel Huigens (editor)
Proton AG
Route de la Galaise 32
CH-1228 Plan-les-Ouates
Switzerland
Email: d.huigens@protonmail.com