

openpgp
Internet-Draft
Intended status: Informational
Expires: 16 March 2026

D. K. Gillmor
ACLU
12 September 2025

OpenPGP External Secret Keys
draft-dkg-openpgp-external-secrets-02

Abstract

This document defines a standard wire format for indicating that the secret component of an OpenPGP asymmetric key is stored externally, for example on a hardware device or other comparable subsystem.

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-external-secrets/>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-dkg-openpgp-external-secrets/>.

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-external-secrets/>.

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 16 March 2026.

Copyright Notice

Copyright (c) 2025 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	3
1.1. Requirements Language	3
1.2. Terminology	4
2. Externally-backed Secret Key Material	4
2.1. Locator Hint	5
2.2. Best Effort Access to External Secret Keys	5
3. Security Considerations	6
4. Usability Considerations	7
4.1. Some Hardware Might Be Unavailable To Some Implementations	7
4.2. Hardware Should Support Multiple Secret Keys	8
4.3. Authorization Challenges	8
4.4. Latency and Error Handling	8
5. IANA Considerations	9
6. References	10
6.1. Normative References	10
6.2. Informative References	10
Appendix A. Historical notes	11
Appendix B. Test vectors	12
B.1. Example Transferable Secret Key	12
B.2. As an External Secret Key	12
Acknowledgements	13
Document History	13
Substantive Changes from draft-dkg-openpgp-external-secrets-01 to draft-dkg-openpgp-external-secrets-02	13
Substantive Changes from draft-dkg-openpgp-external-secrets-00 to draft-dkg-openpgp-external-secrets-01	14
Substantive Changes from draft-dkg-openpgp-hardware-secrets-02 to draft-dkg-openpgp-external-secrets-00	14
Substantive Changes from draft-dkg-openpgp-hardware-secrets-01 to draft-dkg-openpgp-hardware-secrets-02	14

Substantive Changes from draft-dkg-openpgp-hardware-secrets-00 to draft-dkg-openpgp-hardware-secrets-01	14
Author's Address	14

1. Introduction

Some OpenPGP secret key material is held by a hardware device that permits the user to operate the secret key without divulging it explicitly. For example, the [OPENPGP-SMARTCARD] specification is intended specifically for this use. It may also be possible for an OpenPGP implementation to use external secret key material via a standard platform library interface like [TPM].

An OpenPGP Secret Key Packet (see Section 5.5.3 of [RFC9580]) is typically used as part of a Transferable Secret Key (Section 10.2 of [RFC9580]) for interoperability between OpenPGP implementations. An implementation that uses an external secret key needs a standardized way to indicate to another implementation that specific secret key material has been delegated to some external mechanism, like a hardware device.

This document defines a simple mechanism for indicating that a secret key has been delegated to an external mechanism by allocating a codepoint in the "Secret Key Encryption (S2K Usage Octet)" registry (see Section 3.7.2.1 of [RFC9580]).

It also establishes a registry of hints about how to locate the external device, and defines a minimalist "best effort" method for locating external secret keys that is implementation-specific.

This document makes no attempt to specify how an OpenPGP implementation discovers, enumerates, or operates external secret keys, other than to recommend that the hardware or comparable external subsystem should be identifiable by the secret key's corresponding public key material.

1.1. Requirements Language

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.

The key words "PRIVATE USE" and "SPECIFICATION REQUIRED" that appear in this document when used to describe namespace allocation are to be interpreted as described in [RFC8126].

1.2. Terminology

"Secret key" refers to a single cryptographic object, for example the "56 octets of the native secret key" of X448, as described in Section 5.5.5.8 of [RFC9580].

"Public key" likewise refers to a single cryptographic object, for example the "56 octets of the native public key" of X448, as above.

"OpenPGP certificate" or just "certificate" refers to an OpenPGP Transferable Public Key (see Section 10.1 of [RFC9580]).

"External" refers to any cryptographic device or subsystem capable of performing an asymmetric secret key operation using an embedded secret key without divulging the secret to the user. For discoverability, the external mechanism is also expected to be able to produce or be indexed by the public key corresponding to the embedded secret key.

While this document talks about "external" in the abstract as referring to a cryptographic device embedding a single secret key, most actual hardware devices or other cryptographic subsystems will embed and enable the use of multiple secret keys (see Section 4.2).

This document uses the term "authorization" to mean any step, such as providing a PIN, password, proof of biometric identity, button-pushing, etc, that the external subsystem may require for an action.

2. Externally-backed Secret Key Material

An OpenPGP Secret Key packet (Section 5.5.3 of [RFC9580]) indicates that its secret key material is stored in a cryptographic subsystem that is identifiable by public key parameters by setting the S2K usage octet to TBD (252?), known in shorthand as External.

The remainder of the Secret Key packet consists of an optional hint about how the implementation might be able to locate an external subsystem that offers access to the secret key. This locator hint is entirely advisory.

If the locator hint is absent (that is, if there are no bytes in the Secret Key packet following the External S2K usage octet, then the locator hint is known as "best effort" (see Section 2.2).

2.1. Locator Hint

If the locator hint is not empty, then the first octet of the hint describes the structure of the remainder of the hint, according to the "OpenPGP External Secret Key Locator Hints" registry established by this document. That registry is initially empty, with octet values 249-255 (inclusive) reserved for PRIVATE USE. Adding a new entry into that registry in the range 0-248 (inclusive) uses IANA policy SPECIFICATION REQUIRED.

Regardless of what the locator hint says, a consuming implementation MAY use the "best effort" approach to identify an external subsystem that can provide access to the secret key.

A producing implementation that does not know how to provide a meaningful locator hint SHOULD NOT include any trailing data in the rest of such a Secret Key packet.

A consuming implementation that does not understand any particular locator hint SHOULD ignore any trailing data in such a Secret Key packet.

The purpose of the hinting mechanism is to enable optimized access. For example, if an OpenPGP implementation has access to several dozen hardware tokens, and if querying each attached hardware token is expensive, a locator hint can be used to preferentially access a likely token, without probing each token.

This document does not describe any particular scheme of locator hint.

2.2. Best Effort Access to External Secret Keys

When no locator hint is available, or when a consuming implementation does not understand a given locator hint, or when a given locator hint fails to point to a useful device, a consuming implementation uses a "best effort" strategy to identify the external subsystem that provides access to a secret key that matches the corresponding public key material.

Each OpenPGP implementation might support different external subsystems. And, in some installations, the same external subsystem might be identified in different ways (for example, a USB smartcard might be connected to hub 2 at low speed, and in another, the same USB smartcard might be connected to hub 1 at high speed).

In some cases, no external subsystem can be identified that supports access to secret key material that corresponds to the associated public key. Or, the external subsystem might be available, but for whatever reason attempting to use it could fail (for example, the hardware might advertise the availability of the key, but deny access when the implementation tries to use it). In these cases, an OpenPGP implementation that tries to use such an external secret key will fail. The implementation should fail in a similar way to how it might fail if it tried to use a typical software-backed secret key locked with a password, but the password is unavailable to the implementation.

3. Security Considerations

External or hardware-backed secret keys promise several distinct security advantages to the user:

- * Often, the secret key cannot be extracted from the external device, so "kleptography" (the stealing of secret key material) is harder to perform.
- * Some hardware can be moved between machines, enabling secret key portability without expanding the kleptographic attack surface.
- * Some hardware devices offer auditability controls in the form of rate-limiting, user-visible authorization steps (e.g., button-presses or biometric sensors), or tamper-resistant usage counters. Malicious use of a secret key on such a device should be harder, or at least more evident.
- * Some hardware security devices can attest that key material has been generated on-card, thereby signaling that - barring a successful attack on the hardware - no other copy of the private key material exists. Such mechanisms signal that the key holder did not have a chance to mishandle (e.g.: accidentally disclose) the private key material.

However, none of these purported advantages are without caveats.

The hardware itself might actually not resist secret key exfiltration as expected. For example, isolated hardware devices are sometimes easier to attack physically, via temperature or voltage fluctuations (see [VOLTAGE-GLITCHING] and [SMART-CARD-FAULTS]).

In some cases, dedicated cryptographic hardware that generates a secret key internally may have significant flaws (see [ROCA]).

Furthermore, the most sensitive material in the case of decryption is often the cleartext itself, not the secret key material. If the host computer itself is potentially compromised, then kleptographic exfiltration of the secret key material itself is only a small risk. For example, when handling an OpenPGP Encrypted Message, the OpenPGP symmetric session key itself could be exfiltrated, permitting access to the cleartext to anyone without access to the secret key material.

Portability brings with it other risks, including the possibility of abuse by the host software on any of the devices to which the hardware is connected.

Rate-limiting, user-visible authorization steps, and any other form of auditability also suffer from risks related to compromised host operating systems. Few hardware devices are capable of revealing to the user what operations specifically were performed by the device, so even if the user deliberately uses the device to, say, sign an object, the user depends on the host software to feed the correct object to the device's signing capability.

4. Usability Considerations

External secret keys present specific usability challenges for integration with OpenPGP.

4.1. Some Hardware Might Be Unavailable To Some Implementations

This specification gives no hints about how to find the hardware device, and presumes that an implementation will be able to probe available hardware to associate it with the corresponding public key material. In particular, there is no attempt to identify specific hardware or "slots" using identifiers like PKCS #11 URIs ([RFC7512]) or smartcard serial numbers (see Appendix A). This minimalism is deliberate, as it's possible for the same key material to be available on multiple hardware devices, or for a device to be located on one platform with a particular hardware identifier, while on another platform it uses a different hardware identifier.

Not every OpenPGP implementation will be able to talk to every possible hardware device. If an OpenPGP implementation encounters a hardware-backed secret key as indicated with this mechanism, but cannot identify any attached hardware that lists the corresponding secret key material, it should warn the user that the specific key claims to be hardware-backed but the corresponding hardware cannot be found. It may also want to inform the user what categories of hardware devices it is capable of probing, for debugging purposes.

4.2. Hardware Should Support Multiple Secret Keys

Most reasonable OpenPGP configurations require the use of multiple secret keys by a single operator. For example, the user may use one secret key for signing, and another secret key for decryption, and the corresponding public keys of both are contained in the same OpenPGP certificate.

Reasonable hardware SHOULD support embedding and identifying more than one secret key, so that a typical OpenPGP user can rely on a single device for hardware backing.

4.3. Authorization Challenges

Cryptographic hardware can be difficult to use if frequent authorization is required, particularly in circumstances like reading messages in a busy e-mail inbox. This hardware MAY require authorization for each use of the secret key material as a security measure, but considerations should be made for caching authorization.

If the cryptographic hardware requires authorization for listing the corresponding public key material, or for probing whether a given public key matches the device's secret keys, it becomes even more difficult to use the device in regular operation. Hardware SHOULD NOT require authorization for the action of producing the list of corresponding public keys or for probing whether a public key matches the device's secret keys.

If a user has two attached pieces of hardware that both hold the same secret key, and one requires authorization while the other does not, it is reasonable for an implementation to try the one that doesn't require authorization first. Some cryptographic hardware is designed to lock the device on repeated authorization failures (e.g. 3 bad PIN entries locks the device), so this approach reduces the risk of accidental lockout.

4.4. Latency and Error Handling

While hardware-backed secret key operations can be significantly slower than modern computers, and physical affordances like button-presses or NFC tapping can themselves incur delay, an implementation using a hardware-backed secret key should remain responsive to the user. It should indicate when some interaction with the hardware may be required, and it should use a sensible timeout if the hardware device appears to be unresponsive.

A reasonable implementation should surface actionable errors or warnings from the hardware to the user where possible.

5. IANA Considerations

This document asks IANA to make three changes in the "OpenPGP" protocol group.

Add the following row in the "OpenPGP Secret Key Encryption (S2K Usage Octet)" registry:

S2K usage octet	Shorthand	Encryption parameter fields	Encryption	Generate?
TBD (252?)	External	External Locator Hint, see Section 2 of RFC XXX (this document)	no data	Yes

Table 1: Row to add to OpenPGP Secret Key Encryption (S2K Usage Octet) registry

Modify this row of the "OpenPGP Symmetric Key Algorithms" registry:

ID	Algorithm
253, 254, and 255	Reserved to avoid collision with Secret Key Encryption

Table 2: Row to modify in OpenPGP Symmetric Key Algorithms registry

to include TBD (252?) in this reserved codepoint sequence, resulting in the following entry:

ID	Algorithm
TBD (252?), 253, 254, and 255	Reserved to avoid collision with Secret Key Encryption

Table 3: Modified row in OpenPGP Symmetric Key Algorithms registry

Establish a new registry, "OpenPGP External Secret Key Locator Hints", with the following columns and initial range:

ID	Shorthand	Description	Reference
0-191	Unassigned		RFC XXX (this document)
248-255	Private or Experimental Use		RFC XXX (this document)

Table 4: OpenPGP External Secret Key Locator Hints

Assigning a new codepoint to this registry uses SPECIFICATION REQUIRED.

6. References

6.1. 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/rfc/rfc2119>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/rfc/rfc8126>>.
- [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>>.
- [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/rfc/rfc9580>>.

6.2. Informative References

- [GNUPG-SECRET-STUB] Koch, W., "GNU Extensions to the S2K algorithm", 4 July 2023, <[https://dev.gnupg.org/source/gnupg/browse/master/doc/DETAILS;gnupg-2.4.3\\$1511](https://dev.gnupg.org/source/gnupg/browse/master/doc/DETAILS;gnupg-2.4.3$1511)>.

[OPENPGP-SMARTCARD]

Pietig, A., "Functional Specification of the OpenPGP application on ISO Smart Card Operating Systems, Version 3.4.1", 18 March 2020, <<https://gnupg.org/ftp/specs/OpenPGP-smart-card-application-3.4.1.pdf>>.

[RFC7512] Pechanec, J. and D. Moffat, "The PKCS #11 URI Scheme", RFC 7512, DOI 10.17487/RFC7512, April 2015, <<https://www.rfc-editor.org/rfc/rfc7512>>.

[ROCA] Nemec, M., Sys, M., Svenda, P., Klinec, D., and V. Matyas, "The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli", ACM, Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security pp. 1631-1648, DOI 10.1145/3133956.3133969, October 2017, <<https://doi.org/10.1145/3133956.3133969>>.

[SMART-CARD-FAULTS]

Massolino, P. M. C., Ege, B., and L. Batina, "Smart Card Fault Injections with High Temperatures", 15 November 2016, <<http://hdl.handle.net/2117/99293>>.

[TPM] Trusted Computing Group, "Trusted Platform Module Library Specification, Family "2.0", Level 00, Revision 01.59", November 2019, <<https://trustedcomputinggroup.org/resource/tpm-library-specification/>>.

[VOLTAGE-GLITCHING]

Bittner, O., Krachenfels, T., Galauner, A., and J. Seifert, "The Forgotten Threat of Voltage Glitching: A Case Study on Nvidia Tegra X2 SoCs", arXiv, DOI 10.48550/ARXIV.2108.06131, 2021, <<https://doi.org/10.48550/ARXIV.2108.06131>>.

Appendix A. Historical notes

Some OpenPGP implementations make use of private codepoint ranges in the OpenPGP specification within an OpenPGP Transferable Secret Key to indicate that the secret key can be found on a smartcard.

For example, GnuPG uses the private/experimental codepoint 101 in the S2K Specifier registry, along with an embedded trailer with an additional codepoint, plus the serial number of the smartcard (see [GNUPG-SECRET-STUB]).

However, recent versions of that implementation ignore the embedded serial number in favor of scanning available devices for a match of the key material, since some people have multiple cards with the same secret key.

Appendix B. Test vectors

B.1. Example Transferable Secret Key

The OpenPGP Transferable Secret Key used for this example. It includes (unencrypted) private key material for both its primary key and one subkey:

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

```
xVgEZgWtcxYJKwYBBAHaRw8BAQdAlLK6UPQsVHR2ETk1SwVIG3tBmpiEtikYYlCy
1TIiqzYAAQCwm/O5cWsztxbUcwOHycBwszHpD4Oa+fK8XJDxLWH7dRIZzR08aGFy
ZHdhcmUtc2VjcmV0QGV4YW1wbGUub3JnPsKNBBAWCAA1AhkBBQJmBalzAhsDCAsJ
CAcKDQwLBRUKCQgLAhYCFiEEXlP8Tur0WZR+f0I33/i9Uh4OHEkACgkQ3/i9Uh4O
HENryAD8CzH2ajJvASp46ApfI4pLPY57rjBX++d/2FQPRyqGHJUA/RLsNNgxiFYm
K5cjtQe2/DgzWQ7R6PxPC6oa3XM7xPcCx10EZgWtcxIKKwYBBAGXVQEFAQEHE1Y
XOKeaklwG01Yab4xopP9wbulE+pCrPlxQpiFZW5KAWEIBwAA/12uOubAQ5nhf1UF
a5lSQwFLpggB/Spn29qDnSQXOTzIDvPCeAQYFggAIAUCZgWtcwIbDBYhBF5T/E7q
9FmUfn9CN9/4vVieDhxJAAoJEN/4vVieDhxJVTgA/1WaFrKdP3AgL0Ffdooc5XXb
jQsj0uHo6FZSHRI4pchMAQCyJnKQ3RvW/0gm4lJCqImyg2fxWG4hY0N5Q7Rc6Pyz
DQ==
=lYbx
```

-----END PGP PRIVATE KEY BLOCK-----

B.2. As an External Secret Key

The same OpenPGP Transferable Secret Key with the S2K Usage Octet set to 252? (External) for both the Primary Key Packet and the Subkey Packet. This format omits all data following the S2K Usage Octet:

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

```
xTQEZgWtcxYJKwYBBAHaRw8BAQdAlLK6UPQsVHR2ETk1SwVIG3tBmpiEtikYYlCy
1TIiqzb8zR08aGFyZHdhcmUtc2VjcmV0QGV4YW1wbGUub3JnPsKNBBAWCAA1AhkB
BQJmBalzAhsDCAsJCAcKDQwLBRUKCQgLAhYCFiEEXlP8Tur0WZR+f0I33/i9Uh4O
HEkACgkQ3/i9Uh4OHENryAD8CzH2ajJvASp46ApfI4pLPY57rjBX++d/2FQPRyqG
HJUA/RLsNNgxiFYmK5cjtQe2/DgzWQ7R6PxPC6oa3XM7xPcCzkeZgWtcxIKKwYB
BAGXVQEFAQEHE1YXOKeaklwG01Yab4xopP9wbulE+pCrPlxQpiFZW5KAWEIB/zC
eAQYFggAIAUCZgWtcwIbDBYhBF5T/E7q9FmUfn9CN9/4vVieDhxJAAoJEN/4vVie
DhxJVTgA/1WaFrKdP3AgL0Ffdooc5XXbjQsj0uHo6FZSHRI4pchMAQCyJnKQ3RvW
/0gm4lJCqImyg2fxWG4hY0N5Q7Rc6PyzDQ==
=3w/O
```

-----END PGP PRIVATE KEY BLOCK-----

The (primary) Secret-Key Packet of this key looks as follows, in this format:

0000	c5		packet type: Secret-Key Packet
0001	34		packet length
0002	04		version
0003	66 05 ad 73		creation time
0007		16	public-key algorithm: EdDSALegacy
0008	09		curve OID length
0009	2b 06 01 04 01 da 47		curve OID
0010	0f 01		
0012	01 07		EdDSA public key Q MPI length
0014	40 94 b2 ba		EdDSA public key Q MPI
0018	50 f4 2c 54 74 76 11 39		
0020	35 4b 05 48 1b 7b 41 9a		
0028	98 84 b6 29 18 62 50 b2		
0030	d5 32 22 ab 36		
0035		fc	S2K usage octet

Acknowledgements

This work depends on a history of significant work with hardware-backed OpenPGP secret key material, including useful implementations and guidance from many people, including:

- * NIIBE Yutaka
- * Achim Pietig
- * Werner Koch
- * Heiko Schfer
- * Andrew Gallagher

The people acknowledged in this section are not responsible for any proposals, errors, or omissions in this document.

Document History

Substantive Changes from draft-dkg-openpgp-external-secrets-01 to draft-dkg-openpgp-external-secrets-02

- * A device can support indexing (probing for public key material) instead of enumerating available public keys
- * Encourage "best effort" even if an understood locator hint fails to identify a device

- * Private use range is now "Private or Experimental", and aligned with a power of 2

Substantive Changes from draft-dkg-openpgp-external-secrets-00 to draft-dkg-openpgp-external-secrets-01

- * define the external locator hinting mechanism

Substantive Changes from draft-dkg-openpgp-hardware-secrets-02 to draft-dkg-openpgp-external-secrets-00

- * rename from "Hardware-backed" to "External"
- * use RFC9580 instead of I-D.ietf-openpgp-crypto-refresh

Substantive Changes from draft-dkg-openpgp-hardware-secrets-01 to draft-dkg-openpgp-hardware-secrets-02

- * re-format hexdump of test vector secret key packet

Substantive Changes from draft-dkg-openpgp-hardware-secrets-00 to draft-dkg-openpgp-hardware-secrets-01

- * Added test vector for experimentation
- * Mention on-device attestation
- * update OpenPGP card spec reference to 3.4.1

Author's Address

Daniel Kahn Gillmor
American Civil Liberties Union
125 Broad St.
New York, NY, 10004
United States of America
Email: dkg@fifthhorseman.net