



GPL-3-free replacements of GnuPG

1	Contents	
2	Introduction	2
3	Terminology and concepts	3
4	Use cases	3
5	Non-use cases	3
6	Requirements	3
7	Depending components	4
8	apertis-archive-keyring	5
9	APT	5
10	Flatpak	5
11	gmime	5
12	OSTree	6
13	volume-key	6
14	Approach	6
15	RNP	6
16	rPGP	7
17	Sequoia	8
18	golang.org/x/crypto/openpgp	8
19	gpgrv	9
20	Evaluation Report	9
21	Debian upstream discussion	10
22	Recommendations	10
23	Risks	12

24 Introduction

25 In accordance to its [Open Source License Expectations](https://apertis-website-0b3586.pages.apertis.org/policies/license-expectations/)¹, Apertis currently ships
26 a very old version of GnuPG which is still released under the GPL-2.0 terms, before
27 the upstream project switched to GPL-3.0.

28 This is problematic in the long term: the purpose of this document is to investi-
29 gate alternative implementations with licensing conditions that are suitable for
30 Apertis target devices.

31 The use cases for Apertis target images only depend on GnuPG for verification
32 purposes, not for signing or encrypting. This is usually done through the `gpgv`

¹<https://apertis-website-0b3586.pages.apertis.org/policies/license-expectations/>

33 tool or through the `libpgpme` library which invokes the `gpg` tool and interacts with
34 it via the `--with-colons` [machine parsable mode](#)² or the [Assuan](#)³ IPC protocol.

35 Newer `GPL-3`-licensed versions of GnuPG can be provided in the `development`
36 package repository for any additional need outside that do not affect targets.
37 Until `Ed25519` support is officially implemented in APT, the upstream version
38 is imported in Apertis and our infrastructure is reworked to use it rather than
39 OpenPGP signatures, we will need GnuPG to sign and install packages on
40 development images. This does not affect production devices as APT is not
41 meant to be used there.

42 Terminology and concepts

- 43 • **OpenPGP**: The OpenPGP protocol defines standard formats for en-
44 crypted messages, signatures, and certificates for exchanging public keys.
- 45 • **GnuPG**: GnuPG is a complete and free implementation of the OpenPGP
46 standard.

47 Use cases

- 48 • A developer wants to install an additional package on the Apertis APT-
49 based image flashed on their device, and relies on OpenPGP signatures to
50 assert trust in the remote package repositories.
- 51 • A user wants to install a Flatpak application from Flathub, which only
52 provides OpenPGP signatures to assert trust on the provided application
53 bundles.

54 Non-use cases

- 55 • Sending emails encrypted with OpenPGP
- 56 • Creating OpenPGP signatures

57 Requirements

58 The chosen approach to replace GnuPG on targets must:

- 59 • have a license that matches the Apertis [Open Source License Expecta-](#)
60 [tions](#)⁴, including its dependencies
- 61 • provide OpenPGP signature verification support
- 62 • require minimal changes in tools currently depending on GnuPG
- 63 • require minimal non-upstreamable changes

²<https://github.com/gpg/gnupg/blob/master/doc/DETAILS>

³<https://www.gnupg.org/documentation/manuals/assuan/index.html>

⁴<https://apertis-website-0b3586.pages.apertis.org/policies/license-expectations/>

- have an active upstream community
- have a high code quality track

Depending components

GnuPG and the related components are currently used in Apertis for the following packages (based on `apt-rdepends` results):

component	dependent package	source	repository
gnupg	flatpak-tests	flatpak	target
	libgpgme11	gpgme1.0	target
	libvolume-key1	volume-key	target
	ostree-tests	ostree	target
	python-apt		development
	devscripts		development
	gnupg2		development
	jetring		development
libgpgme11	flatpak	flatpak	target
	flatpak-tests	flatpak	target
	libflatpak0	flatpak	target
	gmime-bin	gmime	target
	libgmime-3.0-0	gmime	target
	libgpgmepp6	gpgme1.0	target
	libvolume-key1	volume-key	target
	samba-dsdb-modules	samba	development
gpgv	apertis-archive-keyring		target
	apt		target
	gnupg		target
	devscripts		development
	gpgv2		development

Current packages using GnuPG or gpgv are:

component	dependencies
apertis-archive-keyring	gpgv
apt	gpgv
flatpak	gnupg, libgpgme11
gmime	libgpgme11
ostree	gnupg, libgpgme11(1)
volume-key	gnupg, libgpgme11

(1) Currently `OSTree` in Apertis does not depend on GnuPG as it exclusively uses Ed25519 signatures. However, the reintroduction of OpenPGP signature verifica-

tion support may be requested in the future to be able to verify the provenance and install applications from third-party Flatpak repositories that only provide OpenPGP signatures.

apertis-archive-keyring

This package contains all necessary GnuPG cryptographic keys needed to sign all Apertis archives. The runtime dependency on `gpgv` can be removed with no ill effect.

APT

`gpgv` is used by APT:

- to assert trust on remote package repository indexes
- by `apt-key` which is deprecated⁵ and will be removed
- in build-time tests

Calls to `gpgv` are encapsulated in `ExecGPGV` function located in `apt-pkg/contrib/gpgv.cc`.

At the time this document is written, there's a discussion in Debian mailing list regarding ideas to replace `gpgv` with `sqv`⁶. The emerging long term idea is to have the APT code link to the Sequoia cryptographic library underlying `sqv`, rather than the current approach of invoking an external process.

Flatpak

Flatpak application and library use both `libgpgme11` and `libostree`.

GnuPG is used by Flatpak:

- during development to sign the package and summaries,
- and on target to verify the signatures.

Starting with Apertis v2022dev2, Flatpak is also able to use Ed25519 cryptography.

gmime

GnuPG is used by `gmime` to encrypt, decrypt, sign and verify messages with Multipurpose Internet Mail Extension.

Starting with Apertis v2022dev3, the ability to encrypt, decrypt, sign and verify messages has been disabled in `gmime`.

⁵<https://manpages.debian.org/testing/apt/apt-key.8.en.html>

⁶<https://lists.debian.org/deity/2021/01/msg00088.html>

OSTree

GnuPG is used by OSTree:

- during development to sign the commits,
- and on target to verify the commits.

Current version of OSTree in Apertis is also able to use Ed25519 cryptography.

volume-key

See [Debian manpage](#)⁷.

GnuPG is used by volume-key to encrypt or decrypt the file used to store extracted “secrets” used for volume encryption (for example keys or passphrases).

Starting with Apertis v2022dev3, key escrow support has been disabled in lib-blockdev library, allowing to remove volume-key.

Approach

The following alternative replacements have been considered:

library	License	language	comment
RNP	BSD-2-Clause + BSD-3-Clause + Apache-2.0	C++	
rPGP	Apache-2.0 or MIT	Rust	
Sequoia	GPL-2+	Rust	uses Nettle/GMP
golang.org/x/crypto/openpgp	BSD-3-Clause	Golang	
gpgrv	Apache-2.0 or MIT	Rust	only provides gpg

RNP

<https://github.com/rnp/rnp>

Started in 2017.

RNP originated as an attempt to modernize the NetPGP codebase originally created by Alistair Crooks of NetBSD in 2016. RNP has been heavily rewritten, and carries minimal if any code from the original codebase

Version	# commits	# contributors	CI	gpgv replacement	C API
0.14	2700	31	yes	yes	yes

Used by:

⁷https://manpages.debian.org/buster/volume-key/volume_key.8.en.html

- Thunderbird
- [EnMail](#)⁸ ruby gem

124 **rPGP**

125 <https://github.com/rpgp/rpgp>

126 Started in 2017.

127 Project description from rPGP site:

128 rPGP is the only full Rust implementation of OpenPGP, following
 129 RFC4880 and RFC2440. It offers a minimal low-level API and does
 130 not prescribe trust schemes or key management policies. It fully
 131 supports all functionality required by the Autocrypt 1.1 e-mail en-
 132 cryptation specification.

133 ...

134 rPGP and its RSA dependency got a first independent security re-
 135 view mid 2019. No critical flaws were found. We have fixed and are
 136 fixing some high, medium and low risk ones. We will soon publish
 137 the full review report.

138 Further independent security reviews are upcoming.

139 ...

140 How is rPGP different from Sequoia?

141 Some key differences:

- 142 • rPGP has a more libre license than Sequoia that allows a
 143 broader usage
- 144 • rPGP is a library with a well-defined, relatively small feature-
 145 set where Sequoia also tries to be a replacement for the GPG
 146 command line tool
- 147 • All crypto used in rPGP is implemented in pure Rust, whereas
 148 sequoia uses Nettle, which is implemented in C.

Version	# commits	# contributors	CI	gpgv replacement	C API
0.7.1	334	12	no	no	no, but possible via a Rust shim

149 Used by:

- 150 • [Delta Chat](#), the e-mail based messenger app suite⁹

⁸<https://github.com/riboseinc/enmail>

⁹<https://delta.chat/>

151 Sequoia

152 <https://sequoia-pgp.org/>

153 <https://gitlab.com/sequoia-pgp/sequoia>

154 Started in 2017.

155 Project status:

156 The low-level API is quite feature-complete and can be used encrypt,
157 decrypt, sign, and verify messages. It can create, inspect, and ma-
158 nipulate OpenPGP data on a very low-level.

159 The high-level API is effectively non-existent, though there is some
160 functionality related to key servers and key stores.

161 The foreign function interface provides a C API for some of Sequoia's
162 low- and high-level interfaces, but it is incomplete.

163 There is a mostly feature-complete command-line verification tool
164 for detached messages called 'sqv'.

165 Sequoia uses [Nettle](#)¹⁰ which is dual licensed [LGPL-3.0](#) and [GPL-2.0](#)¹¹, see
166 [COPYING.LESSERv3](#), [COPYINGv3](#), and [COPYINGv2](#) files in the [Nettle](#)
167 [source repository](#)¹². This is compliant with the Apertis [Open Source License](#)
168 [Expectations](#)¹³ since Sequoia itself is licensed under the GPL-2.0 terms.

Version	# commits	# contrib- utors	CI	gpgv re- placement	C API
library: 1.0.0	3948	33	yes	yes	yes
Command line tools: 0.23.0					

169 Used by:

- 170 • Pijul, KIPA, Radicle, see <https://sequoia-pgp.org/projects/>

171 Sequoia is already packaged for Debian bullseye.

172 golang.org/x/crypto/openpgp

173 <https://pkg.go.dev/golang.org/x/crypto/openpgp>

¹⁰<https://git.lysator.liu.se/nettle/nettle>

¹¹<http://www.lysator.liu.se/~nisse/nettle/nettle.html#Copyright>

¹²<https://git.lysator.liu.se/nettle/nettle>

¹³<https://apertis-website-0b3586.pages.apertis.org/policies/license-expectations/>

174 <https://github.com/golang/crypto/tree/master/openpgp>

175 This package is part of the Go crypto package.

Version	# commits	# contributors	CI	gpgv replacement	C API
v0.0.0-20201221181555-ee23a3978ad			no	no	no

176 Used by:

- 177 • Imported by a lot of Go projects, see <https://pkg.go.dev/golang.org/x/crypto/openpgp?tab=importedby>

179 **gpgrv**

180 <https://github.com/FauxFaux/gpgrv>

181 Started in 2017.

182 gpgrv is a Rust library for verifying some types of GPG signatures.

183 It currently able to verify RSA, SHA1, SHA256 and SHA512 signatures.

Version	# commits	# contributors	CI	gpgv replacement	C API
0.3.0 ¹⁴	109	2	no	yes	NA

184 Used by:

- 185 • APT

186 Evaluation Report

187 The `golang.org/x/crypto/openpgp` package only provides a Go interface and would
188 then require substantial effort to be integrated in other places.

189 gpgrv doesn't seem to be actively developed, with the last commit being on
190 August 2020.

191 RNP and Sequoia provide C interfaces and CLI interfaces to encrypt, decrypt,
192 sign or verify files. They have both received a lot of commits, and have many
193 contributors.

194 rPGP does not provide any CLI interface and a C interface would require a Rust
195 shim, but its licensing terms are much more flexible than the Sequoia ones. It
196 is actively developed. but it has fewer commits and contributors than Sequoia.

¹⁴<https://crates.io/crates/gpgrv>

197 Red Hat removed the OpenPGP support from Thunderbird in Red Hat Enter-
198 prise Linux (RHEL), which uses `RNP`, due to not wanting to distribute [Botan](https://botan.randombit.net/)¹⁵,
199 which has inadequate side-channel protection, see Red Hat bugs [1837512](https://bugzilla.redhat.com/show_bug.cgi?id=1837512)¹⁶ and
200 [1886958](https://bugzilla.redhat.com/show_bug.cgi?id=1886958)¹⁷.

201 Debian upstream discussion

202 The Debian APT maintainers are discussing and planning the removal of the
203 dependency on `gpgv` and potentially on OpenPGP as a whole.

204 For the replacement of `gpgv` Debian will likely not use `RNP` due to its Apache
205 License, see [here](#)¹⁸, and expressed some interest in [linking directly to Sequoia](#)¹⁹.

206 However, the Debian APT maintainers expressed concrete interest in [moving](#)
207 [away from OpenPGP altogether](#)²⁰, by changing the [signature mechanism to use](#)
208 [Ed25519 instead](#)²¹.

209 Adopting a solution which is aligned to the upstream goals would save mainte-
210 nance effort in the long term.

211 Recommendations

212 The split between `rPGP` (more permissive license, more limited goals) and Sequoia
213 (more active, GPL-2.0 only) is unfortunate since `rPGP` would be more suitable
214 for us but is also more risky regarding long term maintenance, with Sequoia
215 being more promising in this regard.

216 The problems to be addressed are:

- 217 1. the use of GnuPG via `gpgv` on the target reference images
- 218 2. the use of GnuPG via `libpgpme` on the target reference images

219 For `gpgv` there are two possible approaches:

- 220 1. use `sqv` from Sequoia to replace `gpgv` with basically no changes in the
221 depending components
- 222 2. for GPL-2.0 applications, link to Sequoia directly as the APT maintainers
223 said

224 For `libpgpme` the situation is more complex because the API surface is way
225 bigger and there are no drop-in replacements. In addition Sequoia, by being
226 GPL-2.0 licensed, is not suitable to be directly linked from `GMime`, `OSTree` and
227 `Flatpak` which are LGPL-2.1 and provide libraries that are meant to be linked by

¹⁵<https://botan.randombit.net/>

¹⁶https://bugzilla.redhat.com/show_bug.cgi?id=1837512

¹⁷https://bugzilla.redhat.com/show_bug.cgi?id=1886958

¹⁸<https://lists.debian.org/deity/2021/02/msg00011.html>

¹⁹<https://lists.debian.org/deity/2021/02/msg00004.html>

²⁰<https://lists.debian.org/deity/2021/02/msg00023.html>

²¹<https://wiki.debian.org/Teams/Apt/Spec/AptSign>

228 applications that may be released under licenses incompatible with the GPL-2.0
229 or even proprietary. `rPGP` may be a better choice in this regard.

230 The approach could then be:

- 231 1. ship `sqv` on target images and create a new `sequoia-gpgv` wrapper which
232 sends the correct status codes so that it gets transparently picked up by
233 APT
- 234 2. patch `apertis-archive-keyring` to avoid any runtime dependency on
235 GnuPG
- 236 3. disable OpenPGP support from `OSTree`, replacing it with the use of
237 Ed25519 signatures
 - 238 • this will drop the ability to assert trust when pulling from third
239 party OpenPGP-signed repositories, which has never been a use-case
240 of interest in Apertis
- 241 4. disable OpenPGP support from `Flatpak`, replacing it with the use of
242 Ed25519 signatures
 - 243 • this will drop the ability to assert trust when pulling from third party
244 Flatpak repositories, which is not a use-case of interest for Apertis
245 target devices but at some point is likely to be desirable on the SDK,
246 so we may consider re-introducing GnuPG support only there where
247 the GPL-3 restrictions are not a concern
- 248 5. disable OpenPGP support from `GMime`
 - 249 • this will drop the ability to send/receive encrypted emails when using
250 evolution-data-server, which has never been a use-case of interest in
251 Apertis
- 252 6. disable key escrow support from `libblockdev` so we can drop the `volume-key`
253 package as a whole with its dependency on `libpgpme`
- 254 7. move the `gpgme` source package to the development package repository
- 255 8. move the `gnupg` source package to the development package repository
- 256 9. re-align the `gnupg` source package to Debian

257 With the steps above it would be possible to stop shipping an outdated GnuPG
258 version with limited effort and limited regressions. In particular, disabling
259 OpenPGP support from Flatpak means that it would not be possible to ver-
260 ify the provenance of applications shipped by third-party stores which use
261 OpenPGP like Flathub, and disabling it from GMime would mean that it could
262 not verify or decrypt OpenPGP emails: both regressions have a very limited
263 impact on the Apertis use-cases.

264 In the longer term, other activities can be undertaken to get rid of the down-
265 stream delta introduced above:

- 266 1. engage with the APT upstream maintainers to help them [move away from](#)
267 [OpenPGP signatures](#)²²
- 268 2. engage with OSTree and Flatpak upstream maintainers to dynamically
269 load `libpgpme` that it can be picked up on the SDK where installing GPL-3.0

²²<https://wiki.debian.org/Teams/Apt/Spec/AptSign>

270 components is not an issue and where it can be useful to install applications
271 from third-party store like Flathub
272 3. engage with Flathub people to support `Ed25519` signatures in addition to
273 the OpenPGP ones
274 4. fully re-enable OpenPGP support in the components where it has been
275 disabled by either:
276 5. porting them to use `rPGP` by engaging with the upstream maintainers about
277 implementing minimal Rush shims
278 6. implementing a `libgpgme` backend that invokes Sequoia externally to avoid
279 licensing issues, either by engaging with the `libgpgme` maintainers or the
280 Sequoia maintainers by providing compatibility with the `--with-colons`
281 `machine parsable mode`²³

282 Risks

283 Drop-in reimplementations may not be 100% compatible and thus may cause
284 subtle issues.

²³<https://github.com/gpg/gnupg/blob/master/doc/DETAILS>