



GPL-3-free replacements of GnuPG

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

24 Introduction

25 In accordance to its [Open Source License Expectations](https://jwd.pages.apertis.org/apertis-website/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 gpgrv

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

33 tool or through the `libgpgme` 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://jwd.pages.apertis.org/apertis-website/policies/license-expectations/>

- 64 • have an active upstream community
- 65 • have a high code quality track

66 Depending components

67 GnuPG and the related components are currently used in Apertis for the following
 68 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

69 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

70 (1) Currently `OSTree` in Apertis does not depend on GnuPG as it exclusively uses

71 Ed25519 signatures. However, the reintroduction of OpenPGP signature verifica-
72 tion support may be requested in the future to be able to verify the provenance
73 and install applications from third-party Flatpak repositories that only provide
74 OpenPGP signatures.

75 **apertis-archive-keyring**

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

79 **APT**

80 `gpgv` is used by `APT`:

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

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

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

90 **Flatpak**

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

92 GnuPG is used by `Flatpak`:

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

95 Starting with Apertis v2022dev2, `Flatpak` is also able to use Ed25519 cryptogra-
96 phy.

97 **gmime**

98 GnuPG is used by `gmime` to encrypt, decrypt, sign and verify messages with Mul-
99 tipurpose Internet Mail Extension.

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

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

100 OSTree

101 GnuPG is used by OSTree:

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

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

105 volume-key

106 See [Debian manpage](#)⁷.

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

109 Approach

110 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

111 RNP

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

113 Started in 2017.

114 RNP originated as an attempt to modernize the NetPGP codebase originally
115 created by Alistair Crooks of NetBSD in 2016. RNP has been heavily rewritten,
116 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

117 Used by:

- 118 • Thunderbird

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

- [EnMail](#)⁸ ruby gem

120 rPGP

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

122 Started in 2017.

123 Project description from rPGP site:

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

129 ...

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

134 Further independent security reviews are upcoming.

135 ...

136 How is rPGP different from Sequoia?

137 Some key differences:

- 138 • rPGP has a more libre license than Sequoia that allows a
139 broader usage
- 140 • rPGP is a library with a well-defined, relatively small feature-
141 set where Sequoia also tries to be a replacement for the GPG
142 command line tool
- 143 • All crypto used in rPGP is implemented in pure Rust, whereas
144 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

145 Used by:

- 146 • [Delta Chat, the e-mail based messenger app suite](#)⁹

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

⁹<https://delta.chat/>

147 **Sequoia**

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

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

150 Started in 2017.

151 Project status:

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

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

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

159 There is a mostly feature-complete command-line verification tool
160 for detached messages called ‘sqv’.

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

Version	# commits	# contributors	CI	gpgv replacement	C API
library: 1.0.0	3948	33	yes	yes	yes
Command line tools: 0.23.0					

165 Used by:

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

167 Sequoia is already packaged for Debian bullseye.

168 **golang.org/x/crypto/openpgp**

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

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

171 This package is part of the Go crypto package.

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://jwd.pages.apertis.org/apertis-website/policies/license-expectations/>

Version	# commits	# contributors	CI	gpgv replacement	C API
v0.0.0-20201221181555-ec23a3978ad			no	no	no

172 Used by:

- 173 • Imported by a lot of Go projects, see [https://pkg.go.dev/golang.org/x/](https://pkg.go.dev/golang.org/x/crypto/openpgp?tab=importedby)
174 [crypto/openpgp?tab=importedby](https://pkg.go.dev/golang.org/x/crypto/openpgp?tab=importedby)

175 **gpgrv**

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

177 Started in 2017.

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

179 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

180 Used by:

- 181 • APT

182 Evaluation Report

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

185 `gpgrv` doesn't seem to be actively developed, with the last commit being on
186 August 2020.

187 `RNP` and `sequoia` provide C interfaces and CLI interfaces to encrypt, decrypt,
188 sign or verify files. They have both received a lot of commits, and have many
189 contributors.

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

193 Red Hat removed the OpenPGP support from Thunderbird in Red Hat Enter-
194 prise Linux (RHEL), which uses `RNP`, due to not wanting to distribute `Botan`¹⁵,

¹⁴<https://crates.io/crates/gpgrv/0.3.0>

¹⁵<https://botan.randombit.net/>

195 which has inadequate side-channel protection, see Red Hat bugs [1837512](#)¹⁶ and
196 [1886958](#)¹⁷.

197 **Debian upstream discussion**

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

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

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

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

207 **Recommendations**

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

212 The problems to be addressed are:

- 213 1. the use of GnuPG via `gpgv` on the target reference images
- 214 2. the use of GnuPG via `libgpgme` on the target reference images

215 For `gpgv` there are two possible approaches:

- 216 1. use `sqv` from Sequoia to replace `gpgv` with basically no changes in the
217 depending components
- 218 2. for GPL-2.0 applications, link to Sequoia directly as the APT maintainers
219 said

220 For `libgpgme` the situation is more complex because the API surface is way
221 bigger and there are no drop-in replacements. In addition Sequoia, by being
222 GPL-2.0 licensed, is not suitable to be directly linked from `GMime`, `OSTree` and
223 `Flatpak` which are LGPL-2.1 and provide libraries that are meant to be linked by
224 applications that may be released under licenses incompatible with the GPL-2.0
225 or even proprietary. `rPGP` may be a better choice in this regard.

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>

226 The approach could then be:

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

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

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

- 261 1. engage with the APT upstream maintainers to help them [move away from](#)
262 [OpenPGP signatures](#)²²
- 263 2. engage with OSTree and Flatpak upstream maintainers to dynamically
264 load `libpgpme` that it can be picked up on the SDK where installing GPL-
265 3.0 components is not an issue and where it can be useful to install appli-
266 cations from third-party store like Flathub

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

- 267 3. engage with Flathub people to support `Ed25519` signatures in addition to
268 the OpenPGP ones
- 269 4. fully re-enable OpenPGP support in the components where it has been
270 disabled by either:
- 271 5. porting them to use `rPGP` by engaging with the upstream maintainers about
272 implementing minimal Rush shims
- 273 6. implementing a `libgpgme` backend that invokes Sequoia externally to avoid
274 licensing issues, either by engaging with the `libgpgme` maintainers or the
275 Sequoia maintainers by providing compatibility with the `--with-colons`
276 `machine parsable mode`²³

277 Risks

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

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