Switching from GnuPG to Sequoia Chameleon on Fedora

Stavros Kousidis

Published 2026-09-22
Tested on Fedora 45 · Sequoia Chameleon GnuPG

This guide records a reversible way to use Sequoia Chameleon GnuPG in place of the `gpg` and `gpgv` commands on Fedora. It deliberately keeps the Fedora GnuPG packages installed: Chameleon replaces the command-line implementation, not the complete GnuPG architecture.

The setup is marked **experimental**. Test it with non-critical data before relying on it, especially when desktop mail clients, smart cards or post-quantum certificates are involved.

What changes—and what does not

Sequoia Chameleon GnuPG provides `gpg-sq` and `gpgv-sq`, reimplementations of commonly used `gpg` and `gpgv` functionality using Sequoia PGP.

After the switch:

- OpenPGP message and certificate processing in the selected commands is performed by Sequoia;
- familiar `gpg` command-line syntax remains available for supported operations; and
- the existing GnuPG home directory and public certificates can continue to be used.

However, this is not a complete replacement of GnuPG:

- `gpg-agent` may still manage existing secret keys and perform private-key operations;
- Pinentry may still request and cache passphrases through `gpg-agent`;
- `gpgconf`, smart-card support and related services still come from GnuPG; and
- applications using GPGME may select a fixed engine path instead of the `gpg` found through the shell's `PATH`.

The practical result is a hybrid stack: Sequoia at the `gpg` compatibility layer, with parts of the established GnuPG infrastructure still underneath it.

Record the current state

Before installing anything, record the commands and packages currently in use:

```
command -v gpg gpgv gpg-agent gpgconf
gpg --version
rpm -qf "$(command -v gpg)" "$(command -v gpg-agent)"
gpgconf --list-dirs homedir agent-socket
```

List the public and secret keys that the existing installation sees:

```
gpg --list-keys
gpg --list-secret-keys
```

Back up the OpenPGP configuration and key material before changing the command selection. Preserve file permissions and protect the backup like any other copy of secret-key material.

Install Chameleon alongside GnuPG

Fedora does not currently provide Chameleon as a standard distribution package. One reproducible option is to install the released Rust crate for the current user.

Install the build requirements:

```
sudo dnf install \
  cargo \
  gcc \
  pkgconf-pkg-config \
```

```
  nettle-devel \
  clang-devel \
  sqlite-devel \
  bzip2-devel
```

Then install Chameleon without replacing any Fedora-owned files:

```
cargo install --locked sequoia-chameleon-gnupg
```

The resulting commands normally live below `~/.cargo/bin`:

```
command -v gpg-sq gpgv-sq
gpg-sq --version
gpgv-sq --version
```

If tested RPM packages are available for the intended Fedora release, prefer those over a local Cargo installation. Inspect their contents and dependencies before installing them, and do not remove `gpg-agent`, `gpgconf` or Pinentry as part of the transition.

Test Chameleon before switching

First invoke Chameleon by its explicit name. This leaves the existing `gpg` command untouched:

```
gpg-sq --list-keys
gpg-sq --list-secret-keys
```

Compare the result with the baseline. Differences must be understood before continuing.

Create a small test file and make a detached signature. Replace `YOUR_FINGERPRINT` with the fingerprint of a non-critical test key:

```
printf 'Sequoia Chameleon test\n' > chameleon-test.txt

gpg-sq \
  --local-user YOUR_FINGERPRINT \
  --armor \
  --detach-sign chameleon-test.txt

gpg-sq --verify chameleon-test.txt.asc chameleon-test.txt
```

A passphrase prompt may be displayed by Pinentry. That is expected when the selected secret key is still managed through `gpg-agent`.

Also test an encryption round trip:

```
gpg-sq \
  --recipient YOUR_FINGERPRINT \
  --armor \
  --output chameleon-test.encrypted.asc \
  --encrypt chameleon-test.txt

gpg-sq \
  --output chameleon-test.decrypted.txt \
  --decrypt chameleon-test.encrypted.asc

cmp chameleon-test.txt chameleon-test.decrypted.txt
```

`cmp` produces no output and returns status `0` when both files are identical.

Switch the command for the current user

Do not overwrite `/usr/bin/gpg` or files owned by Fedora packages. Instead, place user-controlled links earlier in the `PATH`:

```
mkdir -p ~/.local/bin
ln -sfn "$(command -v gpg-sq)" ~/.local/bin/gpg
```

```
ln -sfn "$(command -v gpgv-sq)" ~/.local/bin/gpgv
hash -r
```

Confirm which implementation is now selected:

```
type -a gpg gpgv
gpg --version
```

This assumes `~/.local/bin` precedes `/usr/bin` in the login environment. If it does not, adjust the user `PATH` explicitly rather than modifying system files.

Repeat the listing, signature and encryption tests using `gpg` instead of `gpg-sq`.

Confirm the remaining GnuPG components

The compatibility command may still communicate with `gpg-agent`. Inspect the agent socket and process:

```
gpgconf --list-dirs agent-socket
pgrep -a gpg-agent
```

If no process is shown initially, perform a private-key operation and check again. GnuPG normally starts the agent on demand.

This remaining dependency is significant. `gpg-agent` is not merely a graphical prompt: it manages private keys, delegates smart-card operations and coordinates Pinentry. Replacing the packet-processing and command-line layer with Sequoia therefore does not automatically migrate every secret key into a Sequoia-native keystore.

Post-quantum certificates

Chameleon can understand OpenPGP features supported by its Sequoia dependencies, but that alone does not make the complete desktop workflow post-quantum capable. In particular:

- a legacy secret key held by `gpg-agent` can still use the agent fallback;
- a post-quantum secret key requires a backend that can perform its private-key algorithm;
- GPGME and the mail client must accept the certificate and the operations exposed by the selected engine; and
- signing, verification, encryption, decryption and Pinentry interaction must each be tested separately.

Do not infer KMail, Evolution or Claws Mail compatibility from a successful `gpg-sq --list-keys` test. A shell-level switch may not affect the GPGME engine used by those applications, and successful public-certificate parsing does not prove that private-key operations work.

Revert the switch

Remove only the two user-controlled links:

```
rm ~/.local/bin/gpg ~/.local/bin/gpgv
hash -r
```

Verify that Fedora's commands are selected again:

```
command -v gpg gpgv
gpg --version
```

The expected paths are normally `/usr/bin/gpg` and `/usr/bin/gpgv`. The Cargo-installed `gpg-sq` and `gpgv-sq` commands remain available for further side-by-side testing.

Current limitations

Chameleon implements a useful but not feature-complete subset of GnuPG. Trust handling, agent integration, smart cards, application-specific GPGME behavior and newer algorithms require targeted tests. Keep the original Fedora packages installed and retain a tested rollback path.

The important distinction is therefore not simply “GnuPG or Sequoia.” During this migration, both are present: Sequoia processes supported OpenPGP operations through the Chameleon interface, while parts of GnuPG may continue to provide secret-key and desktop integration services.