

# Post-quantum OpenPGP on Fedora 45: shared setup

Stavros Kousidis

**Published** 2026-09-26

**Tested on** Fedora 45 KDE Plasma (KMail and Claws Mail) · Fedora 45 GNOME (Evolution) · Sequoia Chameleon GnuPG 0.13.1 · sequoia-openpgp 2.4.1

This is the shared setup for three client-specific guides:

- KMail
- Claws Mail
- Evolution

The experiment tests hybrid post-quantum OpenPGP mail on Fedora 45 using Sequoia Chameleon as a GnuPG-compatible OpenPGP engine.

Use these constructions:

Signing: ML-DSA-65+Ed25519    OpenPGP algorithm 30  
Encryption: ML-KEM-768+X25519    OpenPGP algorithm 35

A successful UI indication alone is not sufficient evidence. The client-specific guides also inspect the generated OpenPGP packets and independently verify or decrypt the messages.

## Architecture at a glance

KMail

- > QGpgME
- > GPGME++
- > patched GPGME
- > Sequoia Chameleon
- > Sequoia OpenPGP / Keystore

Claws Mail

- > PGP/Core + PGP/MIME
- > patched GPGME
- > Sequoia Chameleon
- > Sequoia OpenPGP / Keystore

Evolution

- > Camel OpenPGP
- > GnuPG-compatible CLI
- > Sequoia Chameleon
- > Sequoia OpenPGP / Keystore

KMail and Claws Mail exercise GPGME and therefore require the experimental RFC 9980 GPGME patch used in this proof of concept. Evolution does not; use Fedora's stock GPGME package for Evolution.

## Install the common components

Enable the COPR and install the Chameleon package:

```
sudo dnf copr enable stavroskousidis/rfc9980-openpgp-poc
```

```
sudo dnf install -y \
    sequoia-chameleon-rfc9980-poc
```

Install Sequoia sq from Fedora:

```
sudo dnf install -y sequoia-sq
```

Check the installed packages:

```
rpm -q \
    sequoia-chameleon-rfc9980-poc \
    sequoia-sq
```

Use Sequoia Chameleon 0.13.1 with sequoia-openpgp 2.4.1. Both Fedora sequoia-sq releases used during testing (1.4.1-1.fc45 and 1.4.1-3.fc45) accept the key-generation syntax below.

## **GPGME: client-specific setup**

### **KMail and Claws Mail**

For KMail and Claws Mail, install the experimental GPGME build from the COPR:

```
sudo dnf upgrade -y \
    --allow-vendor-change \
    --repo="copr:copr.fedorainfracloud.org:stavroskousidis:rfc9980-openpgp-poc" \
    gpgme
```

Confirm that Fedora installs this build:

```
gpgme-2.0.1-6.rfc9980.1.fc45.x86_64
```

The patch adds recognition of the RFC 9980 algorithm IDs needed by the GPGME-based client paths.

### **Evolution**

For Evolution, use Fedora's stock GPGME:

```
gpgme-2.0.1-6.fc45.x86_64
```

If you previously installed the patched build, restore Fedora's package before testing Evolution:

```
sudo dnf copr disable stavroskousidis/rfc9980-openpgp-poc
```

```
sudo dnf distro-sync -y \
    --allow-vendor-change \
    gpgme
```

This operation leaves the Chameleon package installed.

## **Select Chameleon for the desktop session**

In a terminal in the normal desktop session:

```
export PATH="/usr/libexec/rfc9980-openpgp/bin:/usr/bin:/bin"
```

```
command -v gpg
command -v gpgconf
command -v gpgsm
```

```
gpg --version
gpgconf --list-components
```

The expected mapping is:

```
gpg:OpenPGP:/usr/libexec/rfc9980-openpgp/bin/gpg
gpgsm:S/MIME:/usr/bin/gpgsm
```

This intentionally redirects only the OpenPGP CLI path used by the experiment. Fedora's gpgsm remains the S/MIME implementation.

Start the mail client from the same terminal session so that it inherits this PATH.

## **Create a disposable RFC 9980 test certificate**

For an isolated test:

```
mkdir -p "$HOME/rfc9980-mail-test"
cd "$HOME/rfc9980-mail-test"
```

```
sq key generate \  
  --own-key \  
  --userid "RFC9980 PoC <rfc9980-poc@example.invalid>" \  
  --profile rfc9580 \  
  --cipher-suite mldsa65-ed25519 \  
  --encryption-algorithm mlkem768-x25519 \  
  --expiration 1d \  
  --without-password \  
  --output rfc9980-secret.pgp \  
  --rev-cert rfc9980.rev
```

```
sq inspect rfc9980-secret.pgp
```

This command uses one explicit User ID containing both the display name and e-mail address. This is the identity form used for the walkthrough because it integrates cleanly with Evolution's sender-e-mail matching.

Alternatively, generate the certificate with separate name and e-mail User IDs by replacing the `--userid` option with:

```
--name "RFC9980 PoC" \  
--email "rfc9980-poc@example.invalid"
```

Choose one identity form; do not create both certificates for the test. Both forms worked with KMail and Evolution in this proof of concept. Evolution integrated more cleanly with the explicit `--userid` form.

The mail client's account display name is a separate account setting and does not determine how the OpenPGP certificate's User ID is encoded.

The command creates a certificate with an ML-DSA-65+Ed25519 primary/signing structure and an ML-KEM-768+X25519 encryption subkey.

Import it into the Sequoia-backed environment:

```
sq key import rfc9980-secret.pgp
```

Importing the key makes the certificate available to Sequoia, but does not authenticate its User ID. Because this is a self-generated test identity without an external certification path, explicit authorization is required.

This matters because applications may need more than "a suitable key exists": they may require a sufficiently authenticated binding between the sender or recipient identity and that key before permitting cryptographic operations.

Authorize the local test identity using the fingerprint printed by `sq`:

```
sq pki link authorize \  
  --unconstrained \  
  --cert=YOUR_FINGERPRINT \  
  --all
```

Check both interfaces:

```
sq cert list YOUR_FINGERPRINT  
gpg --with-colons --list-keys YOUR_FINGERPRINT  
gpg --with-colons --list-secret-keys YOUR_FINGERPRINT
```

The Chameleon view should expose algorithm 30 for the ML-DSA-65+Ed25519 signing-capable key material and algorithm 35 for the ML-KEM-768+X25519 encryption subkey.

The one-day, passwordless key is intentionally disposable and should not be used as a production identity.

## Test methodology

In each client, create three messages to:

```
rfc9980-poc@example.invalid
```

using:

Unsigned  
Signed only  
Encrypted + signed

Then inspect the signed and encrypted messages independently and confirm:

signed-only signature packet:

```
algorithm 30
MLDSA65_Ed25519
GOODSIG
VALIDSIG
verification exit status 0
```

encrypted-and-signed:

```
version-6 PKESK
algorithm 35
MLKEM768_X25519
DECRYPTION_OKAY
decryption exit status 0
```

inner signature after decryption:

```
algorithm 30
GOODSIG
VALIDSIG
verification exit status 0
```

## Results

| Client           | Desktop              | GPGME              | Verified           |
|------------------|----------------------|--------------------|--------------------|
| KMail 26.08.1    | Fedora 45 KDE Plasma | Experimental patch | Sign ✓ · Encrypt ✓ |
| Claws Mail 4.4.0 | Fedora 45 KDE Plasma | Experimental patch | Sign ✓ · Encrypt ✓ |
| Evolution 3.62.0 | Fedora 45 GNOME      | Fedora stock       | Sign ✓ · Encrypt ✓ |

KMail and Claws Mail use the experimental RFC 9980 GPGME patch. Evolution uses Fedora's stock GPGME package because Camel invokes the GnuPG-compatible Chameleon CLI directly. Independent verification confirms algorithm 30 and 35 packets with GOODSIG, VALIDSIG, DECRYPTION\_OKAY, and exit status 0.

See the client-specific guides for the runtime evidence and message locations.

## Scope

Treat this as an experimental interoperability proof of concept, not as a recommendation to replace Fedora's system OpenPGP stack globally. The COPR packages and compatibility wrapper keep Fedora's standard GnuPG infrastructure intact and allow side-by-side testing without affecting the rest of the system.