Modernizing email encryption

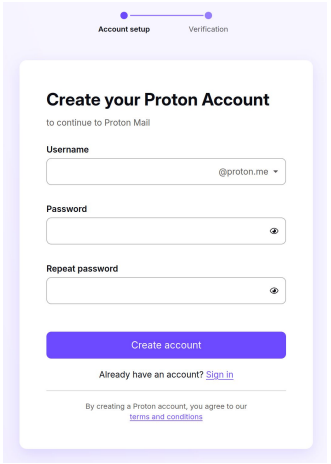
The crypto refresh of OpenPGP

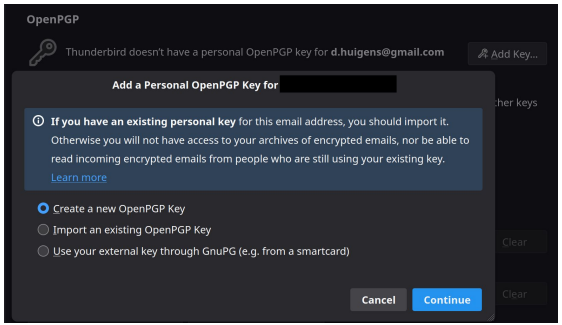
(and other improvements to the ecosystem)

OpenPGP: user experience

Old

New





OpenPGP: libraries

Previously

- GnuPG (C)

Now

- GopenPGP (Go)
- OpenPGP.js (Javascript)
- PGPainless (Java)
- PGPy (Python)
- RNP (C++)
- Sequoia (Rust)

OpenPGP: key distribution

Old

- Manual key distribution

New

- Automatic key distribution :)
  - HTTP Keyserver Protocol
  - Web Key Directory
  - Autocrypt

Modern key lookup: HTTP Keyserver Protocol

GET https://keys.openpgp.org/pks/lookup?op=get&options=mr
&search=john.doe@example.com

Modern key lookup: Web Key Directory

GET https://openpgpkey.example.org/.well-known/openpgpkey/
example.org/hu/ihyath4noz8dsckzjbuyqnh4kbup6h4i?l=john.doe

Old key verification

- In-person exchange / verification
- Key Signing parties
- Web of Trust

New key verification: Key Transparency

- Publish all keys in an append-only log
- Verify your own keys in the log
- Check others’ keys in the log

Key Transparency (cont’d)

- Previous presentation at FOSDEM 2020
- Deployed at Proton
- Whitepaper
- Working group at the IETF

Short history

- RFC 4880 - OpenPGP Message Format: 2007
- RFC 5581 - The Camellia Cipher in OpenPGP: 2009
- RFC 6637 - Elliptic Curve Cryptography (ECC) in OpenPGP: 2012
- draft-koch-eddsa-for-openpgp: 2014
- draft-ietf-openpgp-rfc4880bis: 2016-2020
- draft-ietf-crypto-refresh: 2021-2024

Crypto refresh

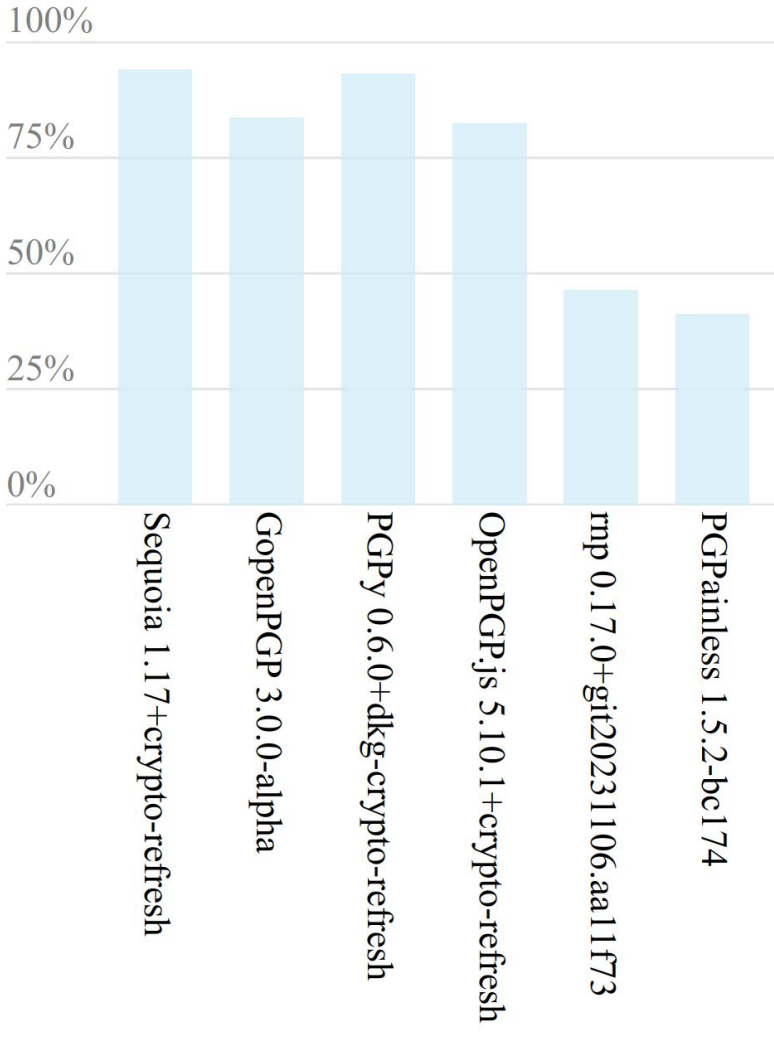
- Merges RFC 5581 (Camellia) and RFC 6637 (ECC) into the main spec
- Adds Curve25519, Curve448, and Brainpool curves
- Adds modern AEAD encryption (OCB, EAX, GCM)
- Adds memory-hard password hashing function (Argon2)
- Deprecates legacy algorithms
- Prevents key overwriting attacks
- Protects against future vulnerabilities
- Adds padding packet

Future improvements

- Post-quantum cryptography
- Forward secrecy
- Key Transparency

Implementations

- Sequoia - 94%
- PGPy - 93%
- GopenPGP - 84%
- OpenPGP.js - 82%
- RNP - 47%
- PGPainless - 41%



OpenPGP vs LibrePGP

OpenPGP

- OCB, EAX, GCM
- Key separation
- Salted hashes
- Memory-hard password hashing function
- Padding

LibrePGP

- OCB
- Signed literal data packet metadata

OpenPGP vs LibrePGP: solutions?

- Compromise?
- Merge?
- Implement both

Conclusion

- We’re dragging OpenPGP into the 21st century
- It’s becoming more and more feasible to build modern E2EE email applications
- Hopefully everyone will implement and use the crypto refresh :)