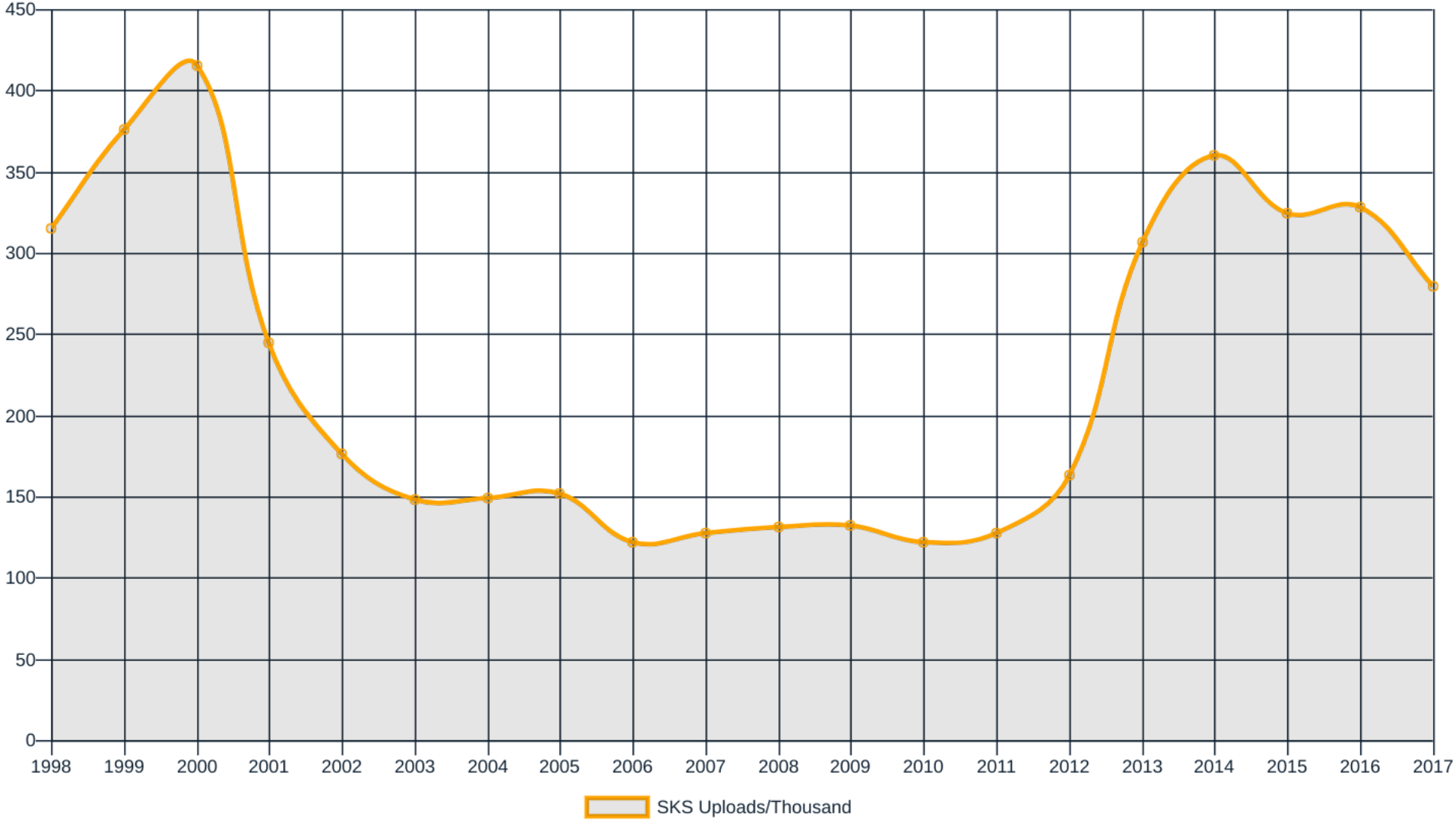
NeoPG

A replacement for GnuPG

Marcus Brinkmann · 34c3 · Leipzig (2017)

OpenPGP is in trouble



OpenPGP Standardization

1998	RFC2440
2007	RFC4880 MAC
2009	RFC5581 Camellia
2012	RFC6637 ECC
2014	*ECC (EdDSA)*
2015	*AEAD, Device Certs*
2017	*SCA, WKD*

2015	IETF WG reopened
2016	*ECC, AEAD, SHA256*
2017	IETF WG closed

"[...] there is not sufficient interest to successfully complete the work of the working group." – IESG Secretary, 11 Nov 2017

20y GnuPG

- GnuPG 1.4
  - *"old, single binary version"*
  - *"support the unsafe PGP-2 keys"*
  - *"no dependencies"* (not even `libgcrypt`)
- GnuPG 2.2
  - depends on `libgpg-error`, `libgcrypt`, `libksba`, `libassuan`, `ntbtls` or `gnutls`, `npth`, `pinentry`

20y libgcrypt

- CVE-2017-0379 - breaks Curve25519 (2016-...)
  - Should have used reference implementation
- CVE-2017-7526 - breaks RSA 1024 (since 2000)
  - Should have looked at OpenSSL 2005-2016
- CVE-2016-6313 - flaw in PRNG (since 1998)
  - Should have used e.g. `HMAC_DRBG` (2012)

20y Key Retrieval

1999 Keyserver (HTTP, HTTPS)

2006 PKA / RFC4398 (DNS)

2015 RFC7929 OPENPGPKEYS (DNS~~SEC~~)

2017 Web Key Directory (DNS+Email+HTTPS)

20y Trust Management

1999 Web of Trust (manual trust delegation)
2002 Web of Trust w/ trust signatures
2015 Trust on first use (TOFU)

NeoPG

- git-style subcommands (and subcommands of subcommands)
- colors!
- gpg2-compatible legacy interface
- single binary (portable apps friendly)
- no system-wide configuration, no daemons, no complicated packaging
- kitchen sink included (hash, compress, armor, random)

Collaboration

- 2-clause BSD license (nobody has time for license wars)
- hosted on GitHub
- pull requests welcome (no contributor agreement needed)

Coding like it's the early 2000's!

- C++11 (gcc >= 4.8, clang, MSVC)
- Heavy use of STL for memory management
- Boost to fill the gaps in STL and abstract platform specific code
- Botan for cryptographic primitives and higher level protocol support

Finally, a library!

- libneopg is the "library for GPG" that never was
- easy high-level interface
- transparent in depth
- all policy decisions replaceable (trust interface, passphrase lookup, etc)
- libgpgme-compatible legacy interface

Focus on Code Quality

- Unit-testing (easy because of libneopg)
- Continuous integration on Linux, MacOS and Windows (Travis, AppVoyeur)
- Fuzzing
- Static code analysis
- Linting, Source Code Formatting (clang-format)

Efficiency

- New key database based on SQLite3.
- Works with large key databases (Debian keyring).
- Efficient programming with high-level abstractions and well-organized components.

Hardware-based security

- Smartcard support out of the box (OpenPGP Card, Gnuk, Yubikey?)
- based on PCSCD (Linux+MacOS, cooperates with other apps)

Beyond the web of trust

- New trust models easy to write.
- keybase.io integration
- Central keyserver that actually verifies email addresses.
- neopg tweet @lambdafu "Do you know yolo-encryption?"

neopg.io

@neopg_