Status Report

Werner Koch

RMLL 2013 — Brussels, July 8, 2013

Outline

Versions

- GnuPG-1: Stable (1.4.13)
- GnuPG-2: Stable (2.0.20)
- Devel: 2.1.0-beta NNN
  - OpenPGP ECC Support
  - OpenPGP keys managed by Gpg-agent

Developers

- Main developers
  - Yutaka Niibe (Smartcards)
  - Jussi Kivilinna (Libgcrypt performance)
  - David Shaw (1.4)
  - me
- Many more are answering questions
- Bug reporters and code reviewers

Legal BS

- ECC patents expired or are worked around.
- (IDEA patent expired)
- Copyright assignments

S/MIME and such

(In case you still want that)

- Complete key management
- CLI allows to create a CSR or self-signed certificate.
- Nice key listings via the CLI and via GPGME.
- Core functions to build a CA should are almost all implemented

The role of the Gpg-Agent

- Takes care of all private keys.
- Generic passphrase cache
- Contacting and scri[pting Gpg-Agent is easy
  - gpg-connect-agent
  - gpgme-tool

The SSH support

- Gpg-agent features a replacement for ssh-agent.
- Latest Windows version has a replacement for Pageant
- Supports smartcards and ECDSA
- I can't live without it.

Our smartcard framework

- Too much to tell
- We don't want to supprt proprietary drivers (aka pkcs#11)
- Supported cards:
  - OpenPGP
  - PKCS#15 cards
  - Some ID cards
  - gnuk
- GPA provides a nice interface

The g13 Prototype

Random access to encrypted data

- Encrypted containers.
- Frontend to disk encryption
- Use your OpenPGP keys and smartcards. (but works also with X.509)

Design:

- Management and Public key encryption
- Different Backends:
  - EncFS (working)
  - dm-crypt (Linux)
  - GELI (BSDs)
  - A LUKS replacement?

GPGME

- The standard API to GnuPG with many language bindings.
- Encryption and signing.
- Configuration interface
- Direct access to the gpg-agent
  - Used for smartcards
- UI-Server interface
  - GPA
  - Kleopatra

GPG secret keys

are finally fully controlled by gpg-agent

- Get rid of lots of code in gpg.
- Merging of secret OpenPGP key is now possible,

The Keybox for OpenPGP

Faster access to OpenPGP keys!

- Used for a decade by gpgsm.
- X.509 and OpenPGP may now be put into the same file.
- Really fast despite its simple structure.
- A keybox with 20000 keys is not a problem anymore (decryption $<$ 100ms with a decent ECC key)
- Indexing can be added.
- Easy to implement a different storage backend (e.g. SQLite)

Pinentry loopback mode

Goal: Easier use of GnuPG-2 on non-desktop boxes.

- loopback passphrase requests.
  $\rightarrow$ Same behaviour as 1.4.
- No need for gpg-preset-passphrase.
- No special pinentry wrappers.

Replacement of GNU Pth

- GNU Pth only used by GnuPG and zhcon.
- Pthreads is now the standard.
- nPth now replaces GNU Pth.

Libgcrypt

Improvements in 1.6 (devel):

- AES-NI (already in 1.5)
- Assembler optimizations for modern CPUs:
  - Better use of AES-NI
  - Use of SSE2
  - Use of AVX
  - For: AES, Camellia, and Serpent,
- Per algorithm code for common modes.
- Overhaul of the ECC code
  - Mainly diue to requests for GNUnet
  - Access to lower levels.

Libgcrypt (cont.)

- Removal of legacy interfaces:
  - `gcry_ac_*`
  - the module reggister subsystem
- Choice of random generator
- Automatic thread initialization.

openpgpjs.org

- Useful for integration into web mailer.
- But do you really want a web mailer?
- Too many new possible attack paths.

Thanks

Many thanks to all the folks who are working on GnuPG, helping others to use is, and waiting for new releases.

`http://gnupg.org`

Thank you for your attention.