Sequoia: A Cool OpenPGP Library

Neal H. Walfield

Delta X, July 21, 2018

https://sequoia-pgp.org/talks/2018-08-introduction
Outline

Introduction

Architectural Overview

Status Quo & Future Directions
Introduction

A new OpenPGP implementation in Rust

Motivation

- GnuPG is hard to modify
  - Code and API grew organically over 21 years
  - Lack of unit tests
  - Tight component coupling
- Many developers unsatisfied with GnuPG’s API
- Rust is memory safe
- GnuPG can’t be used on iOS due to GPL
Introduction

- A new OpenPGP implementation in Rust

- Motivation
  - GnuPG is hard to modify
    - Code and API grew organically over 21 years
    - Lack of unit tests
    - Tight component coupling
  - Many developers unsatisfied with GnuPG’s API
  - Rust is memory safe
  - GnuPG can’t be used on iOS due to GPL
Sequoia’s Goals

▶ Social
▶ Technical
Social Goals

- Inclusive environment
- Free Software
- Community-centered project
  - Development in the open
  - Collaborating with other OpenPGP implementors on design
  - Working with application developers to define API
Technical Goals

- First-class library API, second-class command-line interface
- Friendly API
- Unopinionated low-level API & opinionated high-level API
- Loose component coupling
- Tests, tests, tests, ...
- All modern platforms
- Tight integration with host systems
Who We Are

- Neal, Justus, Kai
  - Former GnuPG developers (2–2.5 years at g10code)
  - At p≡p since Fall 2017

- Funding
  - p≡p (primary)
  - Wau Holland Stiftung (secondary)
  - Actively looking to diversify funding base!
Who We Are

▶ Neal, Justus, Kai
  ▶ Former GnuPG developers (2–2.5 years at g10code)
  ▶ At p≡p since Fall 2017

▶ Funding
  ▶ p≡p (primary)
  ▶ Wau Holland Stiftung (secondary)
  ▶ Actively looking to diversify funding base!
Introduction

Architectural Overview

Status Quo & Future Directions
OpenPGP Crate

- Unopinionated low-level API
- All of RFC 4880
  - Minus the very dangerous bits (MD5, IDEA, ...)
  - Supports streaming
    - Some protection against emission of invalid data
- Intended for advanced OpenPGP use cases
  - Add a signature to an existing message
  - Strip encryption
  - Reencrypt
  - Forensics
  - Analysis
  - etc.
Sequoia Store

- Key database (SQLite based)
  - Public keys
  - Private keys
Public Key Store

- More like a per-domain address book than a PGP keyring
  - Domain: email, software signatures, authentication, …
    - (or per application?)
  - Keys addressed using labels, not user ids
  - Per-domain label ⇒ key mapping, keys shared

- Arbitrary, associated, structured data
  - Useful for implementing trust models

- Keys automatically refreshed in background (à la Parcimonie)
Private Key Store

- Smartcard-like API for using all types of keys
  - Local, Remote, Smartcard, TPM, Trusted Enclave, ...

- One optional password for all local keys

- Automatic key rotation for forward secrecy
  - Mostly compatible with existing OpenPGP implementations
  - (Individual at-rest and transport encryption subkeys)
Sequoia Net

- Accesses keyservers, etc.
- Colocated, not a daemon
Sequoia

- **High-level API**
  - Easy to use
  - Sensible defaults
  - API driven by application needs
  - Minimal manipulation of OpenPGP messages
  - `generate_key`
  - `encrypt_sign`
  - `decrypt_verify`
  - etc.
We maintain important language bindings
Idiomatic interface
C bindings
  Already exist
  Easy to use
Services

- Process separation for security; colocation for reliability
  - Sequoia tries to use a shared service, falls back to colocation
  - When using colocation: synchronization via SQLite
  - IPC protocol: capnproto
Introduction

Architectural Overview

Status Quo & Future Directions
Progress (✓ > 90% done :-)

- **OpenPGP Crate**
  - ✓ Parsing
  - ✓ Serialization
  - ✓ Encryption
  - ✓ Signature Verification
  - ✓ Decryption
  - ✓ Signature Generation
  - △ Key Generation

- **Store**
  - ✓ Public keys
  - △ Private keys
    - △ Local
    - × Smartcards, etc.
    - × Remote
  - × Associated data
  - ✓ Parcimonie

- **Net**
  - ✓ Keyservers
  - × WKD
  - △ Tor support

- **Sequoia Crate**

- **FFI**
  - △ C
  - △ Python
  - × . . .

- **Protocol**
  - △ Forward Secrecy
  - △ Multi-device
  - △ OpenPGP Specification
Ecosystem

- **MUAs**
  - p≡p
  - Enigmail / Thunderbird
  - Delta Chat
  - Leap

- **Infrastructure**
  - Keyserver
  - Kuvert
  - Schleuder

- **Package Managers**
  - sqv (≈ gpgv)
  - Cargo (Rust)

- **Tools**
  - sq split (≈ gpgsplit)
  - pgpdump
  - gpg-sync
  - git
  - wget
Sequoia & Delta Chat

- netpgp-based
  - Incomplete
  - No upstream development
- Uses ca. 10 OpenPGP functions
  - Key generation
  - Sign and encrypt
  - ...

- After 1 day of work... 50% ported!
Sequoia & Delta Chat

- netpgp-based
  - Incomplete
  - No upstream development
- Uses ca. 10 OpenPGP functions
  - Key generation
  - Sign and encrypt
  - ...
- After 1 day of work... 50% ported!
Release Schedule

- sqv and openpgp library this fall
  - ... but no promises :-)
- Further development
  - High-level API
  - Smardcard, etc. support
  - System-specific protection mechanisms
  - Forward secrecy, multi-device support
Release Schedule

- sqv and openpgp library this fall
  - ...but no promises :-) 

- Further development
  - High-level API
  - Smardcard, etc. support
  - System-specific protection mechanisms
  - Forward secrecy, multi-device support
Release Schedule

- sqv and openpgp library this fall
  - ...but no promises :-) 
- Further development
  - High-level API
  - Smardcard, etc. support
  - System-specific protection mechanisms
  - Forward secrecy, multi-device support
Sequoia is a new OpenPGP implementation
User-focused development
Portable & highly integrated
Low-level API is already usable

Join us on...
- irc: #sequoia on Freenode
- mailing list: devel@sequoia-pgp.org
- gitlab:
  gitlab.com/sequoia-pgp/