Stateless OpenPGP Interface

IETF 112 OpenPGP (Nov 2021)

Daniel Kahn Gillmor

draft-dkg-openpgp-stateless-cli

What is `sop`

- Abstract interface for OpenPGP
- "Stateless" -- all arguments are explicitly specified
- Not in-charter for WG

Why `sop`?

- Interop testing
- Clarify concepts
- Encourage best practices

Why "stateless"? Why command line?

- Specify all parts explicitly
- Avoid hidden side effects
- CLI is a "common denominator"

Focus on data management

- key/cert generation
- encrypt/decrypt
- sign/verify

sop Examples

```
sop generate-key "Alice <alice@openpgp.example>" > alice.sec
sop extract-cert < alice.sec > alice.pgp

sop sign --as=text alice.sec < notes.txt > notes.txt.asc
sop verify notes.txt.asc alice.pgp < notes.txt

sop encrypt --sign-with=alice.sec --as=mime bob.pgp\
    < msg.eml > encrypted.asc
sop decrypt alice.sec < ciphertext.asc > cleartext.out
```

Interaction with Crypto Refresh

- Generic interface explicitly does *not* expose algorithm- or version-specific details.
- Can implementation X deal with/interact with wire format object Y?

Missing: Inline Signatures?

- Currently expects and works with detached signatures
- How to deal with bundled message+signature objects?
- See issue 25

Next (1/2): language-specific frameworks

- Java
- Rust
- Python
- C (shared object)?
- Your preferred language?

Next (2/2): Certificate Management

- Merge
- Validate
- Maintain
- Revoke
- Certify
- …?

Recent `sop` Changes

(from -02 to -03: minor changes)

- added `--micalg-out` to `sop sign`
- change from `KEY` to `KEYS`
- new error code `KEY_CANNOT_SIGN`
- `sop version` expanded for more detailed output

Critique, Suggest, Contribute!

https://gitlab.com/dkg/openpgp-stateless-cli