Sequoia PGP

Following a Moral Imperative

Neal H. Walfield <neal@sequoia-pgp.org>

Karakun AG
Basel, Switzerland

November 7, 2023

Outline

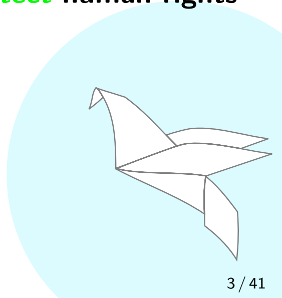
- Human Rights, A Moral Imperative
- A Brief History of PGP
- A Look at OpenPGP
- A Short Introduction to Sequoia

The Internet

- The Internet is wonderful ❤️
- The Internet simplifies abuse 😵‍💫

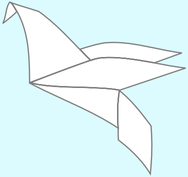
The Internet

- The Internet is wonderful ❤️
- The Internet simplifies abuse 😵‍💫
- The programs that we write are not neutral
- **The programs that we write can *harm* or *protect* human rights**



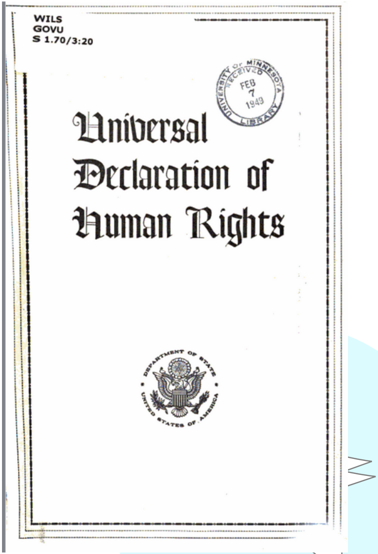
Human Rights

- Human rights are **fundamental**
    - Fundamental means **without compromise**
    - Protecting human rights is a **moral imperative**



What are Human Rights?

- Canonical Reference: UN's *Universal Declaration of Human Rights*
    - Codified 30 freedoms and rights
    - Ratified in 1948

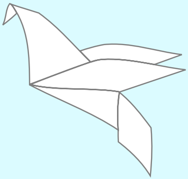


Universal Declaration of Human Rights

Right to Personal Security

Everyone has the right to life, liberty and security of person.

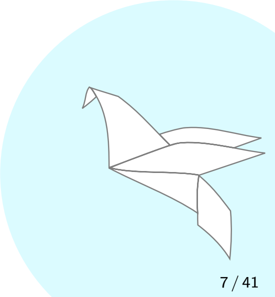
Article 3, *Universal Declaration of Human Rights*



Freedom of Speech

Everyone has **the right to freedom of opinion and expression**; this right includes freedom to hold opinions without interference and to seek, receive and impart information and ideas through any media and regardless of frontiers.

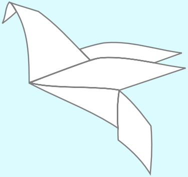
Article 19, *Universal Declaration of Human Rights*



Right to Privacy

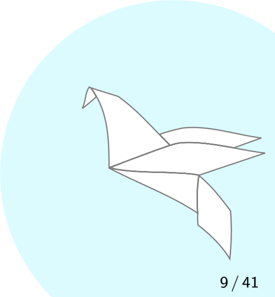
No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

Article 12, *Universal Declaration of Human Rights*



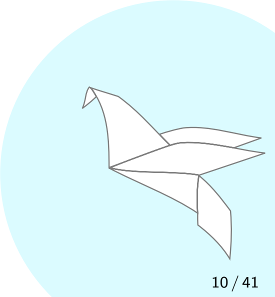
Human Rights

- Security
- Free Expression
- Privacy



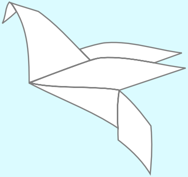
What is Privacy?

- Something to hide



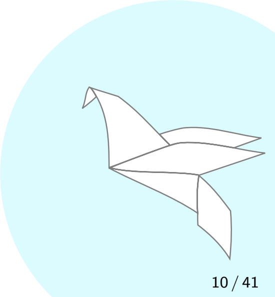
What is Privacy?

- Something to hide?



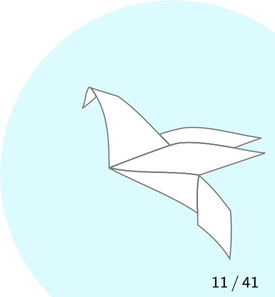
What is Privacy?

- Something to hide?
- No!
- Privacy is a type of **consent**
- Privacy is **control over personal information**



Privacy in Practice

- I may practice ballet 🩰
- I may tell you I practice ballet
- You still have no right to:
    - Watch
    - Spy 👀
- You must have my **informed consent**

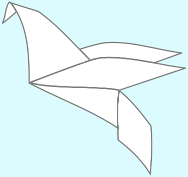


Privacy: Who Cares?

- What happens when someone is watching you?
  - Task becomes performative 🎭
  - Attention is divided
  - Focus is on appearance, not results
  - Afraid to make mistakes

Privacy: Who Cares?

- What happens when someone is watching you?
  - Task becomes performative 🎭
  - Attention is divided
  - Focus is on appearance, not results
  - Afraid to make mistakes
- Examples
  - Exercising 🤸
  - Practicing 👯
  - Learning 🏫
  - Working ⌨️
- Less privacy $\implies$ less experimentation 🥼
- Humans need space for experimentation!



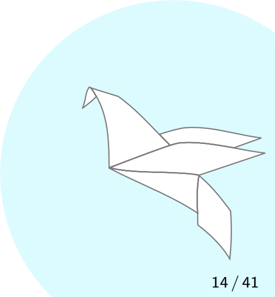
Willful Privacy Violations

- Mass Surveillance
- Surveillance Capitalism
- Dark Patterns
  - Trick users into doing something they wouldn't normally do
  - Trick them into violating their privacy

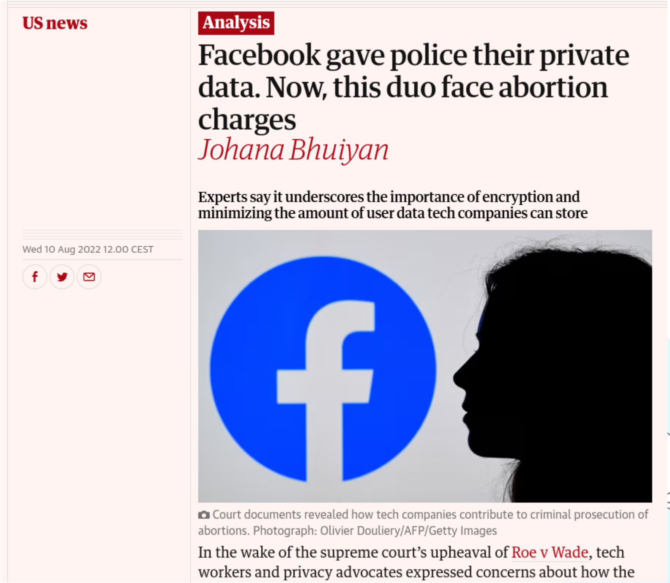
Negligent Privacy Violations

$$\begin{array}{r l} \text{Data Retention} & \text{(Vorratsdatenspeicherung)} \\ + \text{ Data Breaches} & \\ \hline \text{All Data Will Be Public} & \end{array}$$

Even the best get 0wned.



The Harm is Real



US news

Analysis

Facebook gave police their private data. Now, this duo face abortion charges

Johana Bhuiyan

Experts say it underscores the importance of encryption and minimizing the amount of user data tech companies can store

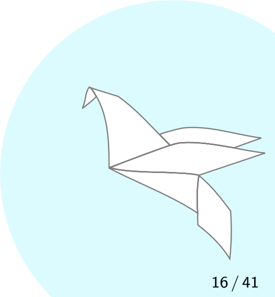
Wed 10 Aug 2022 12.00 CEST

Court documents revealed how tech companies contribute to criminal prosecution of abortions. Photograph: Olivier Douliery/AFP/Getty Images

In the wake of the supreme court's upheaval of Roe v Wade, tech workers and privacy advocates expressed concerns about how the

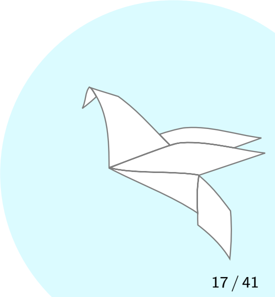
Moral Imperative

- Resist
  - Refuse to violate human rights
  - Refuse to help others violate human rights
  - Educate clients and bosses
  - *Just doing my job* is not an excuse
- Build
  - Design software to protect users' privacy
  - Only collect data that is needed
  - Encrypt, and authenticate



Outline

- Human Rights, A Moral Imperative
- A Brief History of PGP
- A Look at OpenPGP
- A Short Introduction to Sequoia

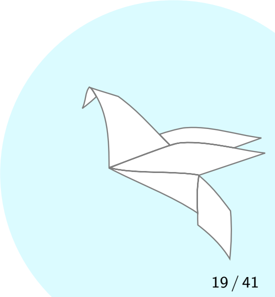


PGP’s Genesis

- Phil Zimmermann, peace activist in the 80s
- Peace groups in an adversarial relationship with the US government
- Need to protect grass root political organisations
- PGP: A tool for cryptographically protecting communication

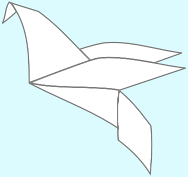
PGP

- First version released in 1991 in US (32 years old!)
- 1993 US criminal investigation for munitions export without a license
    - US only allowed 40-bit encryption
- Workaround: publish a book, users scan code
- US drops case in 1996
- Since then, strong encryption for all



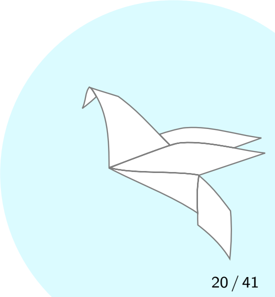
PGP

- First version released in 1991 in US (32 years old!)
- 1993 US criminal investigation for munitions export without a license
  - US only allowed 40-bit encryption
- Workaround: publish a book, users scan code
- US drops case in 1996
- Since then, strong encryption for all
  - ... but constantly under threat
  - EU's Chat Control 2.0



Outline

- Human Rights, A Moral Imperative
- A Brief History of PGP
- A Look at OpenPGP
- A Short Introduction to Sequoia

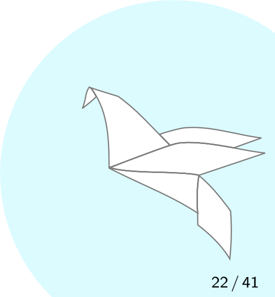


OpenPGP

- IETF standard
- First version: RFC 1991, published in 1996
- Next version: in IETF last call
    - Cryptographic refresh
    - Authenticated encryption
    - Argon2

Actively Developed Free Software Implementations

- GnuPG (C)
- GopenPGP (go)
- OpenPGP.js (javascript)
- PGPainless (Java)
- PGPy (Python)
- RNP (C++)
- rPGP (Rust)
- Sequoia (Rust)

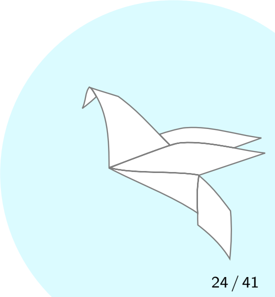


Scope

- Describes a wire format
- Defines encryption, signing, and authentication mechanisms

Two basic data structures

- Messages
- Certificates
- Data structures are made up of packets

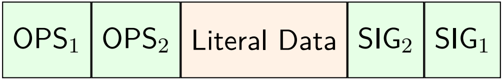


An Encrypted Message

$\mathrm{PKESK}_{r_1}(s)$	$\mathrm{PKESK}_{r_2}(s)$	SEIPD_s

- PKESK: Public Key Encrypted Session Key
    - One per recipient
- SEIPD: Symmetrically Encrypted Integrity Protected Data Packet

A Signed Message



OPS$_1$	OPS$_2$	Literal Data	SIG$_2$	SIG$_1$

- OPS: One Pass Signature
  - Signature metadata
  - Allows for streaming verification
- Literal Data: the actual data
- SIG: A signature
- Signatures nest

The Anatomy of an OpenPGP Certificate

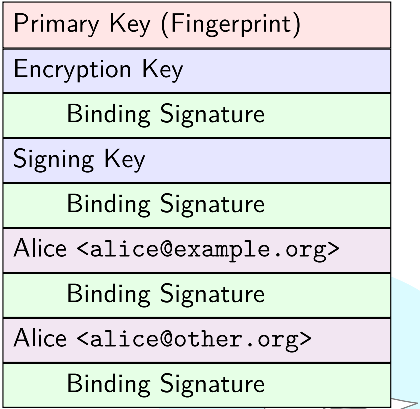
- Public keys
- Identities (User IDs)
- Metadata
  - Expiration
  - Key capabilities
  - User preferences

- Primary Key (Fingerprint)
- Encryption Key
  - Binding Signature
- Signing Key
  - Binding Signature
- Alice `<alice@example.org>`
  - Binding Signature
- Alice `<alice@other.org>`
  - Binding Signature

The Anatomy of an OpenPGP Certificate

- Fingerprint is hash of public key
- Fingerprint uniquely identifies certificate

  8F17777118A33DDA9BA48E62AACB3243630052D9



Primary Key (Fingerprint)

Encryption Key

Binding Signature

Signing Key

Binding Signature

Alice `<alice@example.org>`

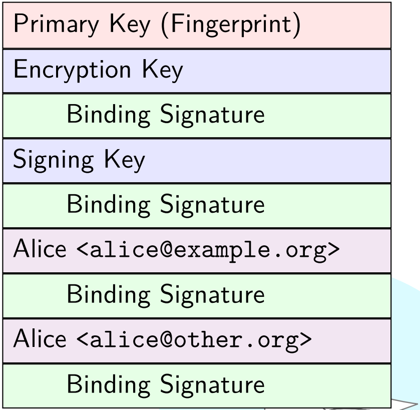
Binding Signature

Alice `<alice@other.org>`

Binding Signature

The Anatomy of an OpenPGP Certificate

- Binding signatures link components
  - Made by primary key
  - Over the primary key and component
- Chain of trust where fingerprint is the trust root
- $\Longrightarrow$ easy to update certificate



Primary Key (Fingerprint)

Encryption Key

Binding Signature

Signing Key

Binding Signature

Alice `<alice@example.org>`

Binding Signature

Alice `<alice@other.org>`

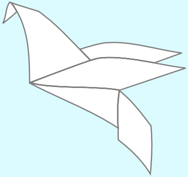
Binding Signature

Public Key Infrastructure: Web of Trust

- Justus certifies that 8F17777118A33DDA9BA48E62AACB3243630052D9 belongs to Neal
  - Justus creates an OpenPGP artifact
  - Artifact can be reasoned about
  - Artifact can be published
- Everyone can act like a certification authority
- Users have their own personal trust roots
- Modes of operation
  - Peer to peer
  - Federated
  - Centralized
- X.509 (Web PKI) is only centralized

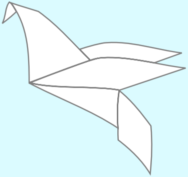
PKI Example: Debian Uploads

- Debian Developers upload their OpenPGP certificate
- Certificates stored in a database
- When uploading a package:
  - System checks signature
  - System checks that issuer is authorized
- Upload can be audited later



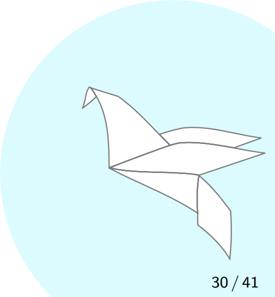
PKI Example: Debian Uploads

- Debian Developers upload their OpenPGP certificate
- Certificates stored in a database
- When uploading a package:
  - System checks signature
  - System checks that issuer is authorized
- Upload can be audited later
- No third-party infrastructure
- Same can be done for a website with encrypted communication
  - Anon.io
  - Facebook



Outline

- Human Rights, A Moral Imperative
- A Brief History of PGP
- A Look at OpenPGP
- A Short Introduction to Sequoia



Sequoia PGP

- Started in 2017
- Founders: Three former GnuPG developers
  - Justus, Kai, Neal
- Primary Sponsors
  - 2017-2023: p≡p Foundation
  - 2023-2024: Sovereign Tech Fund
  - Post 2024: Unknown

Prehistory

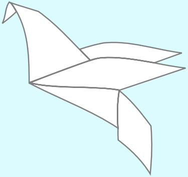
- 2015: Werner Koch hires Neal, Justus & Kai
- Formative period
  - Worked on GnuPG
  - Worked with developers integrating GnuPG
  - Worked with GnuPG users
  - Identified problems
  - Architectural changes too big to do in GnuPG itself
  - Disagreements with Werner about vision
  - Parted ways in Summer 2017

Sequoia's Philosophy

- Not just an OpenPGP implementation
- A project to improve the OpenPGP ecosystem
  - Yes, a new OpenPGP library
  - But also:
    - Improve existing tools
    - Develop new tools
    - Rethink UX paradigms

Approach

- Safety first (Rust)
- Bottom up
- Library first
- Avoid technical debt
- Low-level interfaces are unopinionated, and policy-free
- ...but, secure by default
- Documentation, documentation, documentation



Core Components

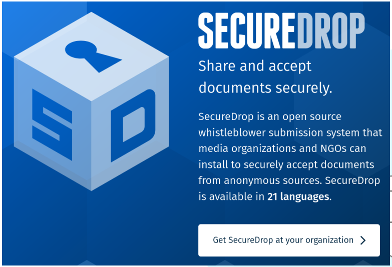
- sequoia-openpgp: Low-level library
- sequoia-net: Networking library
- sequoia-cert-store: Public key store
- sequoia-key-store: Private key store
- sequoia-wot: Web of Trust engine

Sequoia Products

- sq: Command line interface
- Chameleon: gpg replacement
- Octopus: OpenPGP library for Thunderbird
- Hagrid: Software powering `keys.openpgp.org` keyserver
- Sequoia git: software supply chain tool
- OpenPGP Interoperability test suite

A Few Users of Sequoia

- p≡p Engine
  (Key management library)
- RPM Package Manager
- SecureDrop
  (Whistleblower submissions)
- Anon.io
  (Anonymous Email Forwarding)
- Sett (Platform for exchanging medical data in Switzerland)



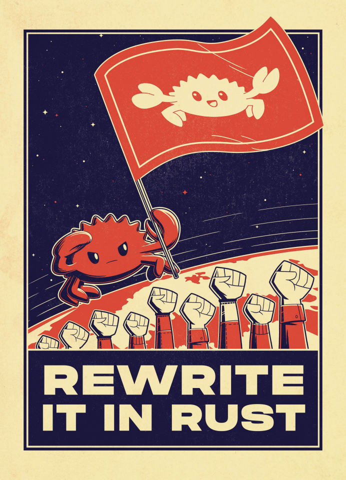
SECUREDROP

Share and accept documents securely.

SecureDrop is an open source whistleblower submission system that media organizations and NGOs can install to securely accept documents from anonymous sources. SecureDrop is available in **21 languages**.

Integrating Sequoia

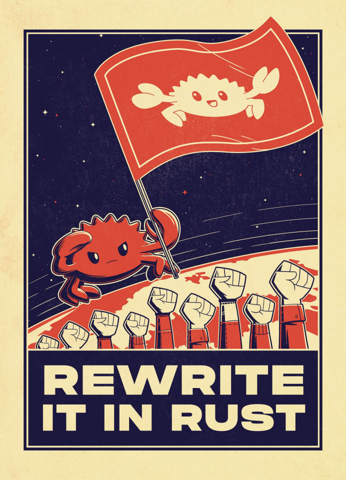
- Two easy ways to integrate Sequoia:
  - `cargo add sequoia-openpgp`
  - Rewrite It In Rust



https://fission.codes/rewrite-in-rust/

Integrating Sequoia

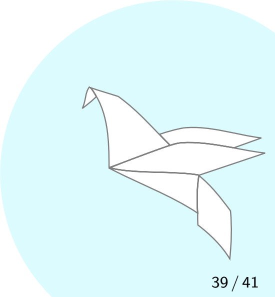
- Two easy ways to integrate Sequoia:
    - `cargo add sequoia-openpgp`
    - ~~Rewrite It In Rust~~ Just kidding 🤣



https://fission.codes/rewrite-in-rust/

Generic Bindings 👎

- Large, low-level API to wrap
- Hard to do in a policy-free manner
- Experience writing a C wrapper was a disaster 😱



Point Solution 👍

- Small Rust library that exports only the needed functionality
- Minimizes impedence mismatches
- Reduces language boundary crossings
- Examples:

p≡p Engine	C	3 727 LOC
rpm	C	2 443 LOC
SecureDrop	Python	411 LOC
Anon.io	PHP	347 LOC

Summary

- We have a *moral imperative* to:
  - *Actively* reject surveillance ✊
  - Collect as little personal data as possible
  - Encrypt the personal data that we collect
- I believe that PGP is a pretty good solution
  - High-level abstractions
  - Good crypto
  - Flexible PKI
  - Active ecosystem
- Sequoia is my preferred implementation 😉