Notable Public Keys

STEVEN BALTAKATEI SANDOVAL

Email: baltakatei@gmail.com

Web: https://reboil.com

Version 0.2.2

Front Matter

The book is available as source code in a `git` repository available at:

https://gitlab.com/baltakatei/npk.git

This book was typeset using **TEXMACS** version `2.1.2`.
Fonts used include **Linux Libertine**.
This book was rendered on `2022-05-17T19:20:11-0700`.

Table of contents

Chapter 1
Trust through Stories

1.1 Summary

This book contains stories about where certain public keys came from and a little about the people who use them.

Some people use public key cryptography to digitally sign their works. They do this so others can prove where copies of such works came from. Usually, digital tools automatically verify these digital signatures so people don't have to manually. However, in order to verify such tools, at some point a person must verify at least one digital signature for themselves.

1.2 Background

As of 2022, most people, if they worry at all about where they download their software from, usually only check that there is a padlock symbol next to the URL in their browser. Thanks to the efforts of LET'S ENCRYPT and other companies promoting use of digital signature technology known as TLS (a.k.a. SSL, HTTPS), most people can rely on that padlock symbol, provided they pay attention to the base domain of the URL (i.e. the "`google.com`" of "`https://mail.google.com`").

TLS works by having a user's web browser come installed with a set of public keys whose private keys are kept secure by IT professionals trusted by governments. These IT people are known as "certificate authorities" (CA). Whenever a webmaster wants to authenticate themselves to visitors to their website, the webmaster may create their own public-private keypair and ask a CA to digitally sign their public key. Then, whenever a visitor's web browser downloads a webpage, the server uses the webmaster's private key to digitally sign the webpage. The web browser can then download the server's public key, see that it is signed by a CA whose public key it already knows about and trusts. This is the cryptographically-secured process that occurs whenever a web browser's padlock symbol indicates a secure TLS connection.

However, for paranoid technically-minded people who want to take precautions against servers being hacked, CA private keys being compromised, or some form of man-in-the-middle attack, sometimes software developers use their own digital certificate systems to authenticate themselves. One such system is OPENPGP. Instead of relying upon CAs trusted by governments, each software developer is their own CA. Unlike with TLS and web browsers, users who wish to verify digital signatures on programs made by such developers must have some trusted means of identifying and acquiring the developers' public keys. With OPENPGP, although it is possible in theory to create and maintain a "Web of Trust" by having key owners regularly sign eachothers' keys based upon their personal relationships with one another, in practice this method of establishing trust is outcompeted by the simplicity of using TLS; if the stakes of misidentifying a team member on a project are high enough, it is much simpler to simply meet in-person.

1.3 Purpose

That said, the purpose of this book is to provide you, reader, a means of identifying public keys used to sign notable software and data. Notability is defined and applied as in WIKIPEDIA: it is a test to determine whether a chapter about an entity's public keys should be included. Where potential for confusion exists around the identity of a notable entity that maintains a public key, this book should identify that key.

This document is a tertiary reference meant to paint a narrative about how and by whom a public key is used. Often public keys are secured by individual software developers and used to sign commits made in their version control systems. Some public keys are used by an individual but to represent an entire company or project. Although most public keys in this book are OPENPGP keys compatible with the GNUPG program, some public keys may use other systems or protocols such as those in TLS certificates, SSH key pairs, or cryptocurrency wallets[1.3.1], as long as they are notable.

This book started as a set of personal notes I began maintaining in 2018 to help me verify software packages that I use. In 2021 I decided to share these notes in book-form with the help of the GNU TEXMACS typesetting program (mainly for its indexing and open-source nature). As of 2022, the method of verification of key notability (me, A0A2 95AB DC34 69C9, scanning the web for fingerprints and keys of programs I use) is not scalable. However, this book uses the GIT version control system and lives in a GITLAB repository so additional collaborators (you) could help this book grow.

1.3.1. E.g.: The address of the first spendable Bitcoin. See https://chainflyer.bitflyer.com/Block/Height/1 .

Chapter 2
List of Public Keys

Each section in this chapter contains a story about a person or organization that uses a public-private key pair. Each story consists of some brief background information, a history of notable events, and public key information. Public keys are usually identified through key fingerprints. Links to public keys are made available where possible[2.0.1].

2.0.1. A set of minimal copies of GNUPG public keys is available in the GIT repository of this book in `ref/pgp_keys/`. File names contain the full 160-bit hexadecimal fingerprint.

2.1 BITCOIN CORE

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.1.1 Background

BITCOIN CORE[2.1.1] is the "reference implementation" of the BITCOIN protocol. It is maintained by a group of people who have become known as the BITCOIN CORE developers.

Early in the blockchain's history, the software that verified transactions against balances of previous transactions was a WINDOWS executable known as BITCOIN. The initial release of this software was by an entity that called themselves SATOSHI NAKAMOTO. Satoshi later gave up the code maintainer role of the project. The person who subsequently gained control was a person named GAVIN ANDRESEN. The software was rebranded from BITCOIN to BITCOIN CORE at version *0.9.0*.[2.1.2] A developer named WLADIMIR J. VAN DER LAAN became owner of the OPENPGP signing keys of the reference implementation starting at version *0.9.3*. VAN DER LAAN originally used a personal key (7481 0B01 2346 C9A6) to sign binaries but later created a dedicated key (90C8 019E 36C2 E964) to sign binaries. In 2021, binaries were instead signed by a group of people each with their own personal key.

There exist various dubious theories regarding PGP key use by SATOSHI NAKAMOTO.[2.1.3] The most likely candidate (18C0 9E86 5EC9 48A1) is one signed by BITCOIN CORE developers PETER TODD (7FAB 1142 67E4 FA04) and WLADIMIR J. VAN DER LAAN (7481 0B01 2346 C9A6).

2.1.2 History

2011-08-24. Creation date of VAN DER LAAN's personal signing key 7481 0B01 2346 C9A6.

2011-12-15. Creation date of ANDRESEN's dedicated code signing key 29D9 EE6B 1FC7 30C1.

2013-03-23. Earliest snapshot of the https://bitcoin.org website on the INTERNET ARCHIVE.[2.1.4] It is a redirect to https://bitcoin.org/en .

2013-04-11. Earliest snapshot of the https://bitcoincore.org website on the INTERNET ARCHIVE.[2.1.5]

2013-07-27. Earliest snapshot of main GITHUB repository at https://github.com/bitcoin/bitcoin on the INTERNET ARCHIVE.[2.1.6]

2014-03-19. The reference client rebranded from BITCOIN to BITCOIN CORE.

2014-04-08. GAVIN ANDRESEN steps down as lead developer. Hands over role to WLADIMIR J. VAN DER LAAN.[2.1.7] ANDRESEN maintains commit privileges to the GITHUB repository.

2015-06-24. Creation date of VAN DER LAAN's dedicated code signing key 90C8 019E 36C2 E964.

2016-05-02. GAVIN ANDRESEN's commit privileges revoked by other BITCOIN CORE developers after ANDRESEN published a blog post claiming CRAIG WRIGHT was SATOSHI NAKAMOTO.[2.1.8]

2021-09-13. BITCOIN CORE version *0.22.0* published with change to how binary releases are signed. Releases now signed by several individuals[2.1.9] instead of VAN DER LAAN's dedicated code signing key 90C8 019E 36C2 E964.[2.1.10]

2.1.1. Main website: https://bitcoincore.org/ .

2.1.2. See https://bitcoin.org/en/release/v0.9.0#rebranding-to-bitcoin-core .

2.1.3. See https://www.vice.com/en/article/jpgq3y/satoshis-pgp-keys-are-probably-backdated-and-point-to-a-hoax .

2.1.4. See https://web.archive.org/web/20130323195546/http://bitcoin.org/en .

2.1.5. See https://web.archive.org/web/20130411033932/http://bitcoincore.org/ .

2.1.6. See https://web.archive.org/web/20130727135658/https://github.com/bitcoin/bitcoin .

2.1.7. See https://www.coindesk.com/gavin-andresen-steps-bitcoins-lead-developer .

2.1.8. See https://twitter.com/peterktodd/status/727078284345917441 , https://laanwj.github.io/2016/05/06/hostility-scams-and-moving-forward.html , https://www.bbc.com/news/technology-36202904 , and https://www.theguardian.com/technology/2016/may/06/bitcoin-project-blocks-out-gavin-andresen-over-satoshi-nakamoto-claims .

2.1.9. For list of fingerprints, see https://web.archive.org/web/20210725054312/https://github.com/bitcoin/bitcoin/blob/master/contrib/builder-keys/keys.txt .

2.1.10. See https://web.archive.org/web/20210926105351/https://bitcoincore.org/en/download/ ; specifically "Linux verification instructions".

2.1.3 Public Key Details

2.1.3.1 Binary Signing Keys (v0.22.0–)

Since the release of BITCOIN CORE version *0.22.0* on 2019-09-13, the `SHA256SUMS` file available at https://bitcoincore.org/en/download has been split into two files:

`SHA256SUMS`. Contains hashes of the binary executable files.

`SHA256SUMS.asc`. Contains multiple detached signtaures from different public keys.

For version *0.22.0*, the fingerprints and primary UIDs of these signatures are in Table 2.1.1.

Fingerprint	UID Name
099B AD16 3C70 FBFA	Will Clark
0A41 BDC3 F4FA FF1C	Aaron Clauson (sipsorcery)
7481 0B01 2346 C9A6	Wladimir J. van der Laan
796C 4109 063D 4EAF	Jon Atack
4101 0811 2E7E A81F	Hennadii Stepanov (GitHub key)
8E42 5659 3F17 7720	Oliver Gugger
944D 35F9 AC3D B76A	Michael Ford (bitcoin-otc)
2EBB 056F D847 F8A7	Stephan Oeste (it)
C37B 1C1D 44C7 86EE	Duncan Dean
E13F C145 CD3F 4304	Antoine Poinsot
D7CC 770B 81FD 22A8	Ben Carman
1756 5732 E08E 5E41	Andrew Chow (Official New Key)

Table 2.1.1. Fingerprints of OPENPGP keys that signed hashes of BITCOIN CORE *v0.22.0* release files.

2.1.3.2 Binary Signing Key (v0.11.0–v0.21.2) (90C8 019E 36C2 E964)

This key[2.1.11], owned by WLADIMIR J. VAN DER LAAN, was used to sign BITCOIN CORE releases between versions *0.11.0* and *0.21.2*.

```
pub   rsa4096/0x90C8019E36C2E964 2015-06-24 [SC] [expired: 2022-02-10]
      Key fingerprint = 01EA 5486 DE18 A882 D4C2  6845 90C8 019E 36C2 E964
uid                  [ expired] Wladimir J. van der Laan (Bitcoin Core binary...) <69ed48c5>
```

2.1.3.3 Binary Signing Key (v0.9.3–v0.10.2) (7481 0B01 2346 C9A6)

WLADIMIR VAN DER LAAN used his personal key[2.1.12] to sign BITCOIN versions v0.9.3–v0.10.2.

```
pub   rsa2048/0x74810B012346C9A6 2011-08-24 [SC] [expires: 2027-02-08]
      Key fingerprint = 71A3 B167 3540 5025 D447  E8F2 7481 0B01 2346 C9A6
uid                   [ unknown] Wladimir J. van der Laan <b64e04a4>
uid                   [ unknown] Wladimir J. van der Laan <69ed48c5>
uid                   [ unknown] Wladimir J. van der Laan <c7357718>
sub   rsa2048/0x69B4C4CDC628F8F9 2017-05-17 [A] [expires: 2027-02-08]
      Key fingerprint = 53D9 74DA 0BAF FF22 B3A5  FB5C 69B4 C4CD C628 F8F9
sub   rsa2048/0xF69705ED890DE427 2011-08-24 [E]
      Key fingerprint = D01B 5D68 0154 44D2 71DA  D33F F697 05ED 890D E427
sub   rsa2048/0x1E4AED62986CD25D 2017-05-17 [S] [expires: 2027-02-08]
      Key fingerprint = 9DEA E0DC 7063 249F B054  7468 1E4A ED62 986C D25D
```

2.1.3.4 Binary Signing Key (v0.8.6–v0.9.2.1) (29D9 EE6B 1FC7 30C1)

GAVIN ANDRESEN used this dedicated code-signing key[2.1.13] to sign BITCOIN versions v0.8.6–v0.9.2.1. As of 2021-07-19, these versions and their signatures are available at https://bitcoincore.org/bin/insecure/ .

2.1.11. See https://reboil.com/res/2021/txt/20210719_90C8019E36C2E964..bitcoin_vanderlaan.asc

2.1.12. See https://reboil.com/res/2021/txt/20210719_74810B012346C9A6..bitcoin_vanderlaan.asc

2.1.13. See https://reboil.com/res/2021/txt/20210719_29D9EE6B1FC730C1..bitcoin_andresen.asc

```
pub   rsa4096/0x29D9EE6B1FC730C1 2011-12-15 [SC]
      Key fingerprint = 2664 6D99 CBAE C9B8 1982  EF60 29D9 EE6B 1FC7 30C1
uid                   [ unknown] Gavin Andresen (CODE SIGNING KEY) <84976526>
sub   rsa4096/0x1B7BFB457BF6E212 2013-11-01 [S]
      Key fingerprint = 3D22 F497 DEAE D078 18A2  219A 1B7B FB45 7BF6 E212
sub   rsa4096/0x36E924A98E30B3ED 2011-12-15 [E]
      Key fingerprint = DC9F CD02 AB25 5459 FDA8  7469 36E9 24A9 8E30 B3ED
```

2.1.3.5 SATOSHI NAKAMOTO (18C0 9E86 5EC9 48A1)

The dsa1024 algorithm this key[2.1.14] uses is considered weak by the the NIST standard SP800-57 Part 1 Revision 5: *Recommendation for Key management*.[2.1.15] The key offers only 80 bits of security against the possibility of impersonation via a brute force attack. Nevertheless, this key has a signature of BITCOIN CORE developer PETER TODD (7FAB 1142 67E4 FA04) dated 2013-10-12. TODD also committed the full fingerprint in a BITCOIN FOUNDATION document on 2013-04-26[2.1.16]. This key also has a signature of BITCOIN CORE maintainer VLADIMIR J. VAN DER LAAN's personal key (7481 0B01 2346 C9A6) dated 2013-05-10, albeit revoked on 2016-05-02.

```
pub   dsa1024/0x18C09E865EC948A1 2008-10-30 [SC]
      Key fingerprint = DE4E FCA3 E1AB 9E41 CE96  CECB 18C0 9E86 5EC9 48A1
uid                   [ unknown] Satoshi Nakamoto <6bab5993>
sig 3        0x18C09E865EC948A1 2008-10-30  Satoshi Nakamoto <6bab5993>
sig          0x74810B012346C9A6 2013-05-10  Wladimir J. van der Laan <c7357718>
sig 1        0x7FAB114267E4FA04 2013-10-12  Peter Todd <13866f62>
rev          0x74810B012346C9A6 2016-05-02  Wladimir J. van der Laan <c7357718>
      reason for revocation: User ID is no longer valid
sub   elg2048/0xCF1857E6D6AAA69F 2008-10-30 [E]
      Key fingerprint = EA4E 9C90 7F72 21B0 B37D  0940 CF18 57E6 D6AA A69F
```

2.1.14. See https://reboil.com/res/2021/txt/20210719_18C09E865EC948A1..bitcoin_nakamoto.asc

2.1.15. See https://doi.org/10.6028/NIST.SP.800-57pt1r5 , table 2, page 54. dsa1024 keys have only offer 80 bits of security against brute force attacks.

2.1.16. See https://github.com/pmlaw/The-Bitcoin-Foundation-Legal-Repo/commit/fb70771a9927e04ebe5ae33c46ba6589a9703e40 .

2.2 CRYPTOMATOR

Last updated 2022-03-15 by STEVEN BALTAKATEI SANDOVAL.

2.2.1 Background

CRYPTOMATOR[2.2.1] is a cross-platform file storage privacy application. It permits storing files on a third-party file storage services (e.g. DROPBOX) in encrypted form and accessible to the user as a virtual mountable drive. In other words, CRYPTOMATOR acts as an encryption layer between a user and a file storage service. Compiled binary releases are available for WINDOWS, MACOS, LINUX, ANDROID, and IOS[2.2.2].

As of 2021-12-22, the latest version of CRYPTOMATOR is version *1.6.5 (Hotfix)* available on GITHUB.[2.2.3] Judging from commit signatures of the GITHUB repository[2.2.4], the main developers appear to be SEBASTIAN STENZEL (667B 866E A824 0A09) ARMIN SCHRENK (748E 55D5 1F5B 3FBC), and TOBIAS HAGEMANN (69CE FAD5 1959 8989).

2.2.2 History

2015-01-01. First snapshot of https://cryptomator.org captured on the INTERNET ARCHIVE.[2.2.5] Signature of latest version of CRYPTOMATOR (*1.4.11*) uses PGP key 509C 9D63 34C8 0F11.[2.2.6]

2018-06-17. Binary signing PGP key 509C 9D63 34C8 0F11 published as GITHUB gist.[2.2.7] Key used to sign CRYPTOMATOR versions prior to *1.5.8*.

2020-09-01. Binary signing PGP key 615D 449F E6E6 A235 published as GITHUB gist.[2.2.8] Key used to sign CRYPTOMATOR version *1.5.8* onward (as of 2021-12-22).

2020-09-02. Old binary signing PGP key 615D 449F E6E6 A235 signed by new PGP key 509C 9D63 34C8 0F11.[2.2.9] Notice of revocation of old key and signing of new key by old key posted in GITHUB issue thread.[2.2.10]

2.2.3 Public Key Details

2.2.3.1 Binary signing key (−v1.5.7) (509C 9D63 34C8 0F11)

PGP key used to sign compiled binary releases of CRYPTOMATOR prior to version *1.5.8*.

```
pub   rsa4096/0x509C9D6334C80F11 2016-06-24 [SC] [revoked: 2020-08-18]
      Key fingerprint = 5054 3A3D A4B1 DB81 DA3E  79CB 509C 9D63 34C8 0F11
uid                   [ revoked] Cryptobot (Release Manager) <d0228975>
```

2.2.1. Main website: https://cryptomator.org/ .

2.2.2. See https://cryptomator.org/downloads/ .

2.2.3. See https://github.com/cryptomator/cryptomator/releases/tag/1.6.5 .

2.2.4. See https://github.com/cryptomator/cryptomator .

2.2.5. See https://web.archive.org/web/20150101033915/http://cryptomator.org/ .

2.2.6. Signature file at: https://web.archive.org/web/20210502041159/https://dl.bintray.com/cryptomator/cryptomator/1.4.11/cryptomator-1.4.11-x86_64.AppImage.asc . Signed file at: https://web.archive.org/web/20210502115653/https://dl.bintray.com/cryptomator/cryptomator/1.4.11/cryptomator-1.4.11-x86_64.AppImage .

2.2.7. See https://gist.github.com/cryptobot/8ccf8fd686d0c2d8381b69126bb3f2f8/9fdeef62bddf9edf7b73f61f42423f1f123d3218 .

2.2.8. See https://gist.github.com/cryptobot/211111cf092037490275f39d408f461a/1a8e133a1d7e6ae4eb2bcc0830e4567393e5162a .

2.2.9. See https://gist.github.com/cryptobot/211111cf092037490275f39d408f461a/d416c6f0d35506116436cbe2f872baa217f3f72a . Verify with `$ gpg --import` and `$ gpg --list-signatures` to show the signature (**highlighted**):

```
pub   rsa4096/0x615D449FE6E6A235 2020-08-18 [SC] [expires: 2031-01-01]
      Key fingerprint = 5811 7AFA 1F85 B3EE C154  677D 615D 449F E6E6 A235
uid                   [ unknown] Cryptobot <d0228975>
sig 3        0x615D449FE6E6A235 2020-08-18  Cryptobot <d0228975>
sig          0x509C9D6334C80F11 2020-09-02  Cryptobot (Release Manager) <d0228975>
```

2.2.10. See https://github.com/cryptomator/cryptomator.github.io/issues/25#issuecomment-685308263 .

2.2.3.2 Binary signing key (v1.5.8–) (615D 449F E6E6 A235)

PGP key used to sign compiled binary releases of CRYPTOMATOR after version *1.5.8* (as of 2021-12-22). The key and fingerprint are available on the Linux download page.[2.2.11]

```
pub   rsa4096/0x615D449FE6E6A235 2020-08-18 [SC] [expires: 2031-01-01]
      Key fingerprint = 5811 7AFA 1F85 B3EE C154  677D 615D 449F E6E6 A235
uid                   [ unknown] Cryptobot <d0228975>
```

2.2.11. See https://web.archive.org/web/20211222094028/https://cryptomator.org/downloads/linux/thanks/# .

2.3 DEBIAN

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.3.1 Background

DEBIAN[2.3.1] is a free operating system from which many GNU/LINUX systems are derived. Such derived systems include UBUNTU, TAILS (see 2.12), KALI LINUX, and others.

DEBIAN is maintained by an association of developers who use GNUPG keys to sign announcements of software they contribute in order to protect against forgeries. A `git` repository containing GNUPG keyrings of DEBIAN keys is available at `https://salsa.debian.org/debian-keyring/keyring` or by installation of the `debian-keyring` package[2.3.2] within a DEBIAN system.

The DEBIAN PROJECT was founded in 1993 by IAN ASHLEY MURDOCK. Various individuals have led the project since.[2.3.3] As of 2021-09-25, the latest release of the oeprating system is called "DEBIAN 11 (BULLSEYE)".

2.3.2 History

1993-08-16. The DEBIAN PROJECT officially founded by IAN ASHLEY MURDOCK.

1999-01-30. Creation date of the Debian CD signing key `7C3B 7970 88C7 C1F7`.

2000-09-16. Creation date of SANTIAGO GARCIA MANTINAN's key `72FD C205 F6A3 2A8E`.

2004-06-20. Creation date of DANIEL BAUMANN's key `F82E 5CC0 4B2B 2B9E`.

2009-10-03. Creation date of the Debian CD signing key `9880 21A9 64E6 EA7D`.

2011-01-05. Creation date of the Debian CD signing key `DA87 E80D 6294 BE9B`.

2014-04-15. Creation date of the Debian Testing CDs Automatic Signing Key `4246 8F40 09EA 8AC3`.

2.3.3 Public Key Details

2.3.3.1 Installation Image Signature Keys

The DEBIAN website makes available images of the operating system that can be installed onto and executed from removable media such as Compact Discs (CD), Digital Versatile Disc (DVD), and Universal Serial Bus (USB) storage devices. A set of GNUPG public key fingerprints have been listed on the `debian.org` website at `https://debian.org/CD/verify` . Table 2.3.1 summarizes the creation dates, long IDs, and availabilities of these keys. Full fingerprints and other information may be found in section 2.3.3.2.

2.3.1. Main website: `https://www.debian.org` .

2.3.2. See `https://tracker.debian.org/pkg/debian-keyring`

2.3.3. For a list of DEBIAN Project Leaders, see `https://www.debian.org/doc/manuals/project-history/leaders` .

Date	Long ID	Description	Available	Link
1999-01-30	7C3B 7970 88C7 C1F7	Debian CD signing key	2011–2015 [2.3.4]	[2.3.5]
2000-09-16	72FD C205 F6A3 2A8E	Santiago Garcia Mantinan	2011–2015 [2.3.4]	[2.3.6]
2004-06-20	F82E 5CC0 4B2B 2B9E	Daniel Baumann	2011–2015 [2.3.4]	
2009-05-21	39BE 2D72 5CEE 3195	Daniel Baumann	2011–2015 [2.3.4]	
2009-10-03	9880 21A9 64E6 EA7D	Debian CD signing key	2011–2021 [2.3.4]	
2011-01-05	DA87 E80D 6294 BE9B	Debian CD signing key	2011–2021 [2.3.4]	[2.3.5]
2011-03-09	6F95 B499 6CA7 B5A6	Debian Live Signing Key	2012–2015 [2.3.7]	
2013-05-06	510A D6B9 AD11 CF6A	Debian Live Signing Key	2013–2015 [2.3.8]	
2014-01-03	1239 00F2 A9B2 6DF5	Live Systems Project	2014–2015 [2.3.9]	
2014-04-15	4246 8F40 09EA 8AC3	Debian Testing CDs Automatic Signing Key	2014–2022 [2.3.10]	

Table 2.3.1. A list of keys used to sign DEBIAN installation images. Keys identified from INTERNET ARCHIVE snapshots of https://debian.org/CD/verify .

2.3.4. See https://web.archive.org/web/20110413065857/http://www.debian.org/CD/verify .

2.3.7. See https://web.archive.org/web/20120815030316/http://www.debian.org:80/CD/verify .

2.3.8. See https://web.archive.org/web/20130813130619/http://www.debian.org/CD/verify .

2.3.9. See https://web.archive.org/web/20140410065231/http://www.debian.org/CD/verify .

2.3.10. See https://web.archive.org/web/20140528012106/https://www.debian.org/CD/verify .

2.3.5. Public key available at https://web.archive.org/web/20210928205206/https://www.einval.com/~steve/pgp/ .

2.3.6. Public key available at https://web.archive.org/web/20210928220426/https://reboil.com/res/2021/txt/20210928_72FDC205F6A32A8E..debian_manty.asc .

2.3.3.2 Verbose key details

Key 1999-01-30 (7C3B 7970 88C7 C1F7)

A 1024-bit DSA key that is the earliest dated key for signing Debian CDs mentioned at https://debian.org/CD/verify according to the INTERNET ARCHIVE [2.3.11]. Mention of this key was removed from that page by the end of 2015. A copy of this key can be found at the personal website of STEVE MCINTYRE, a debian developer.[2.3.12]

```
pub   dsa1024/0x7C3B797088C7C1F7 1999-01-30 [SCA] [revoked: 2017-01-11]
      Key fingerprint = AC65 6D79 E362 32CF 77BB  B0E8 7C3B 7970 88C7 C1F7
uid                   [ revoked] Steve McIntyre <313467b6>
uid                   [ revoked] Steve McIntyre <8b5d9233>
uid                   [ revoked] Steve McIntyre <032c14f6>
uid                   [ revoked] Debian CD signing key <047be9f4>
```

Key 2000-09-16 (72FD C205 F6A3 2A8E)

A 1024-bit DSA key listed as being a signing key for Debian CD images as of 2011 at https://debian.org/CD/verify according to the INTERNET ARCHIVE. Mention of this key was removed from that page by the end of 2015. A copy of this key was archived from the pgp.mit.edu keyserver.[2.3.13] This 1024-bit DSA key was deprecated in favor of a 4096-bit RSA key with fingerprint B868 8CA3 D876 D5A3 in a signed blog post at blog.manty.net .[2.3.14]

```
pub   dsa1024/0x72FDC205F6A32A8E 2000-09-16 [SCA]
      Key fingerprint = 3F0A 12FC 0B55 A917 D791  82D3 72FD C205 F6A3 2A8E
uid                   [ unknown] Santiago Garcia Mantinan (manty) <65783682>
uid                   [ unknown] Santiago Garcia Mantinan (manty) <ad3a4b28>
uid                   [ unknown] Santiago Garcia Mantinan (manty) <99df45ac>
sub   elg1024/0x8F802C268D0EB704 2000-09-16 [E]
      Key fingerprint = 0481 1B16 A1BC E82B E985  26B7 8F80 2C26 8D0E B704
```

2.3.11. See https://web.archive.org/web/20110413065857/http://www.debian.org/CD/verify.

2.3.12. Key 7C3B 7970 88C7 C1F7 is available at https://web.archive.org/web/20210928205229/https://www.einval.com/~steve/pgp/7C3B797088C7C1F7.asc .

2.3.13. Key 72FD C205 F6A3 2A8E is available at https://web.archive.org/web/20210928220426/https://reboil.com/res/2021/txt/20210928_72FDC205F6A32A8E..debian_manty.asc .

2.3.14. Key transition statement available at https://web.archive.org/web/20150614033612/http://blog.manty.net/2014/12/transitioning-from-0xf6a32a8e-to.html . To verify, use `gpg --import` command on text copied from between the `<listing>` tags. A copy of this text is also archived at https://web.archive.org/web/20210928222521/https://reboil.com/res/2021/txt/20210928..72FDC205F6A32A8E_to_B8688CA3D876D5A3_pgp_transition_statement.txt .

Key 2004-06-20 (F82E 5CC0 4B2B 2B9E)

A key listed as being a signing key for Debian CD images as of 2011 at https://debian.org/CD/verify according to the INTERNET ARCHIVE. Mention of this key was removed from that page by the end of 2015.

```
pub   dsa1024/0xF82E5CC04B2B2B9E 2004-06-20 [SC] [expired: 2015-01-01]
      Key fingerprint = 709F 54E4 ECF3 1956 2332  6AE3 F82E 5CC0 4B2B 2B9E
uid                   [ expired] Daniel Baumann <09d45987>
```

Key 2009-05-21 (39BE 2D72 5CEE 3195)

A key listed as being a signing key for Debian CD images as of 2011 at https://debian.org/CD/verify according to the INTERNET ARCHIVE.

```
pub   rsa4096/0x39BE2D725CEE3195 2009-05-21 [SC]
      Key fingerprint = D2FB 633A DDC2 0485 CBCE  6D12 39BE 2D72 5CEE 3195
uid                   [ unknown] Daniel Baumann <58c7ad3d>
uid                   [ unknown] Daniel Baumann <1d8e385c>
uid                   [ unknown] Daniel Baumann <612d81a0>
uid                   [ unknown] Daniel Baumann <2260b338>
uid                   [ unknown] Daniel Baumann <8f3d581a>
uid                   [ unknown] Daniel Baumann <24ba8964>
uid                   [ unknown] Daniel Baumann <09d45987>
uid                   [ unknown] Daniel Baumann <5923ca47>
sub   rsa4096/0x2E86B0C2E7D77F65 2009-05-21 [E]
      Key fingerprint = 205A 272D 2838 238C 3058  C278 2E86 B0C2 E7D7 7F65
```

Key 2009-10-03 (9880 21A9 64E6 EA7D)

A key listed as being a signing key for Debian CD images as of 2011 at https://debian.org/CD/verify according to the INTERNET ARCHIVE.

```
pub   rsa4096/0x988021A964E6EA7D 2009-10-03 [SC]
      Key fingerprint = 1046 0DAD 7616 5AD8 1FBC  0CE9 9880 21A9 64E6 EA7D
uid                   [ unknown] Debian CD signing key <047be9f4>
```

Key 2011-01-05 (DA87 E80D 6294 BE9B)

A key listed as being a signing key for Debian CD images as of 2011 at https://debian.org/CD/verify according to the INTERNET ARCHIVE.

```
pub   rsa4096/0xDA87E80D6294BE9B 2011-01-05 [SC]
      Key fingerprint = DF9B 9C49 EAA9 2984 3258  9D76 DA87 E80D 6294 BE9B
uid                   [ unknown] Debian CD signing key <047be9f4>
sub   rsa4096/0x642A5AC311CD9819 2011-01-05 [E]
      Key fingerprint = 47A8 EA16 451B F5C9 B691  5C64 642A 5AC3 11CD 9819
```

Key 2011-03-09 (6F95 B499 6CA7 B5A6)

This key was mentioned at https://debian.org/CD/verify at the end of 2012, according to the INTERNET ARCHIVE.

```
pub   rsa4096/0x6F95B4996CA7B5A6 2011-03-09 [SC] [expired: 2021-02-01]
      Key fingerprint = 696F 95F0 88E4 D359 947F  7AEB 6F95 B499 6CA7 B5A6
uid                   [ expired] Debian Live Signing Key <4719e7fe>
```

Key 2013-05-06 (510A D6B9 AD11 CF6A)

This key was mentioned at https://debian.org/CD/verify at the end of 2013, according to the INTERNET ARCHIVE.

```
pub   rsa4096/0x510AD6B9AD11CF6A 2013-05-06 [SC]
      Key fingerprint = 1E4F 435C 4E9A 42B3 D9DF  BE3A 510A D6B9 AD11 CF6A
uid                   [ unknown] Debian Live Signing Key (2013) <4719e7fe>
sub   rsa4096/0x4E534D59B72E3E00 2013-05-06 [E]
      Key fingerprint = 407B 17AD 8CD1 B9D3 6891  262B 4E53 4D59 B72E 3E00
```

Key 2014-01-03 (1239 00F2 A9B2 6DF5)

This key was mentioned at https://debian.org/CD/verify at the end of 2014, according to the INTERNET ARCHIVE.

```
pub   rsa4096/0x123900F2A9B26DF5 2014-01-03 [SC] [expires: 2025-01-01]
      Key fingerprint = 8A36 A2E8 91A5 C2A9 0DEB  7A8B 1239 00F2 A9B2 6DF5
uid                   [ unknown] Live Systems Project <4719e7fe>
sub   rsa4096/0xA1A89023D0125917 2014-01-03 [E] [expires: 2025-01-01]
      Key fingerprint = A9E3 8E70 E798 7E03 285E  D9C2 A1A8 9023 D012 5917
```

Key 2014-04-15 (4246 8F40 09EA 8AC3)

This key was mentioned at https://debian.org/CD/verify at the end of 2014, according to the INTERNET ARCHIVE.

```
pub   rsa4096/0x42468F4009EA8AC3 2014-04-15 [SC]
      Key fingerprint = F41D 3034 2F35 4669 5F65  C669 4246 8F40 09EA 8AC3
uid                   [ unknown] Debian Testing CDs Automatic Signing Key <047be9f4>
sub   rsa4096/0x0C5470136BD05CFB 2014-04-15 [E]
      Key fingerprint = AEE9 9CA7 0C3E C4B3 1C75  2843 0C54 7013 6BD0 5CFB
```

2.4 ELECTRUM

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.4.1 Background

ELECTRUM[2.4.1] is a BITCOIN wallet program first published by THOMAS VOEGTLIN (2BD5 824B 7F94 70E6) in 2013.[2.4.2] The program permits creating and digitally signing and verifying BITCOIN transactions as well as messages. In contrast to BITCOIN CORE, the main ELECTRUM program is a "thin client" designed to require less CPU power and storage capacity in order to be able to run on mobile devices such as ANDROID phones. It therefore requires trusting an ELECTRUM server to provide it with current transaction data from the BITCOIN blockchain. One such server is ELECTRUMX[2.4.3], maintained by SOMBERNIGHT (E7B7 48CD AF5E 5ED9).

2.4.2 History

2011-12-14. Early I.A. mention of THOMAS VOEGTLIN's 2BD5 824B 7F94 70E6 key on http://bitcoin-otc.com .[2.4.4]

2013-01-09. Earliest I.A. snapshot of https://electrum.org/download.html available.[2.4.5]

2014-09-13. Earliest I.A. snapshot of https://github.com/spesmilo/electrum available.[2.4.6]

2018-03-31. Creation date of SOMBERNIGHT's E7B7 48CD AF5E 5ED9 key.

2.4.3 Public Key Details

2.4.3.1 ELECTRUM client release signing key (v1.7–) (2BD5 824B 7F94 70E6)

OPENPGP key used by THOMAS VOEGTLIN (a.k.a. "ThomasV") to sign ELECTRUM (a BITCOIN wallet program) releases since as early as version *1.7* in 2013[2.4.7]. A copy of the key can be downloaded from the ELECTRUM website.[2.4.8]

```
pub   rsa4096/0x2BD5824B7F9470E6 2011-06-15 [SC]
      Key fingerprint = 6694 D8DE 7BE8 EE56 31BE  D950 2BD5 824B 7F94 70E6
uid                   [ unknown] Thomas Voegtlin (https://electrum.org) <cab2cac8>
uid                   [ unknown] ThomasV <45704e55>
uid                   [ unknown] Thomas Voegtlin <45704e55>
sub   rsa4096/0x1A25C4602021CD84 2011-06-15 [E]
      Key fingerprint = 0A40 B328 1212 5B08 FCBF  90EC 1A25 C460 2021 CD84
```

2.4.3.2 ELECTRUMX server commit key (E7B7 48CD AF5E 5ED9)

OPENPGP key used by SOMBERNIGHT (a.k.a. "ghost43"[2.4.9]) to sign ELECTRUMX (an ELECTRUM server) commits on GITHUB[2.4.10] as of 2022-03-10. Note, this key is different from another key (CA9E EEC4 3DF9 11DC) used by SOMBERNIGHT to sign ELECTRUM releases.

```
pub   rsa4096/0xE7B748CDAF5E5ED9 2018-03-31 [SC]
      Key fingerprint = 4AD6 4339 DFA0 5E20 B3F6  AD51 E7B7 48CD AF5E 5ED9
uid                   [ unknown] SomberNight <dbf39566>
sub   rsa4096/0xB38EBEDF07698B05 2018-03-31 [E]
      Key fingerprint = 9145 6DB8 FDF2 16A9 3C37  F0A9 B38E BEDF 0769 8B05
sub   rsa4096/0xB33B5F232C6271E9 2018-03-31 [S]
      Key fingerprint = 6D7A 2116 DA90 9E00 AC21  108B B33B 5F23 2C62 71E9
```

2.4.1. Main website: https://electrum.org .
2.4.2. THOMAS VOEGTLIN's GITHUB page: https://github.com/ecdsa .
2.4.3. See https://github.com/spesmilo/electrumx/ .
2.4.4. See https://web.archive.org/web/20111214231353/http://bitcoin-otc.com/viewgpg.php?nick=ThomasV .
2.4.5. See https://web.archive.org/web/20130109155542/http://electrum.org/download.html .
2.4.6. See https://web.archive.org/web/20140913002252/https://github.com/spesmilo/electrum .
2.4.7. See https://web.archive.org/web/20130320184418/http://electrum.org/download.html .
2.4.8. See https://raw.githubusercontent.com/spesmilo/electrum/master/pubkeys/ThomasV.asc .
2.4.9. SOMBERNIGHT's GITHUB page: https://github.com/SomberNight .
2.4.10. See https://github.com/spesmilo/electrumx/commit/914938264e5621ea8980be6d3e69964e7f219d16 .

2.5 F-DROID

Last updated 2022-04-11 by STEVEN BALTAKATEI SANDOVAL.

2.5.1 Background

F-DROID[2.5.1] is an ANDROID app and repository that differentiates itself from the GOOGLE PLAY repository and app by providing only apps with free and open-source software (FOSS) licenses. The F-DROID client permits searching and downloading of FOSS apps in the F-DROID repository as well as downloading and verifying updates for these apps.

The F-DROID project was founded by CIARAN GULTNIEKS in 2010. As of 2022, various individuals[2.5.2] maintain the client but the GIT repository tags are digitally signed using the OPENPGP key of HANS-CHRISTOPH STEINER (E9E2 8DEA 00AA 5556).

2.5.2 History

2010-10-25. Earliest I.A. snapshot of https://f-droid.org .

2014-04-25. Creation date of the ANDROID client binary release signing key (41E7 044E 1DBA 2E89).

2015-10-31. Creation date of the ANDROID client GIT repository signing key (E9E2 8DEA 00AA 5556).

2017-10-17. Earliest snapshot of https://f-droid.org that links to a PGP signature for the client APK.

2017-12-20. Early I.A. mention of the 41E7 044E 1DBA 2E89 and E9E2 8DEA 00AA 5556 signing keys.[2.5.3]

2.5.3 Public Key Details

2.5.3.1 ANDROID client binary release signing key (2017–) (41E7 044E 1DBA 2E89)

OPENPGP key used to sign official binary release of the F-DROID Android client since at least as far back as 2017.[2.5.4]

```
pub   rsa4096/0x41E7044E1DBA2E89 2014-04-25 [C]
      Key fingerprint = 37D2 C987 89D8 3119 4839  4E3E 41E7 044E 1DBA 2E89
uid                   [ unknown] F-Droid <25ef2ab4>
sub   rsa3072/0x7A029E54DD5DCE7A 2014-04-25 [S] [expires: 2026-04-25]
      Key fingerprint = 802A 9799 0161 1234 6E1F  EFF4 7A02 9E54 DD5D CE7A
sub   rsa3072/0x5DCCB667F9BF9046 2014-04-25 [E] [expires: 2026-04-25]
      Key fingerprint = 2622 41A6 D092 3980 0242  8AE6 5DCC B667 F9BF 9046
```

2.5.3.2 ANDROID client GIT repository signing key (2015–) (E9E2 8DEA 00AA 5556)

OPENPGP key used to sign GIT tags of the F-DROID client source code repository since 2015. The key is owned by by HANS-CHRISTOPH STEINER.

```
pub   rsa4096/0xE9E28DEA00AA5556 2015-10-31 [C]
      Key fingerprint = EE66 20C7 136B 0D2C 456C  0A4D E9E2 8DEA 00AA 5556
uid                   [ unknown] Hans-Christoph Steiner <90782c16>
uid                   [ unknown] Hans-Christoph Steiner <e4e6522f>
uid                   [ unknown] Hans-Christoph Steiner <a055eb1b>
uid                   [ unknown] [jpeg image of size 5375]
sub   rsa2048/0x044051F83354A28B 2015-10-31 [E] [expires: 2023-05-26]
      Key fingerprint = F9CC 6FE0 12C9 26C7 37F3  74FC 0440 51F8 3354 A28B
sub   rsa2048/0x3E177817BA1B9BFA 2015-10-31 [S] [expires: 2023-05-26]
      Key fingerprint = 9722 39DB E686 99F5 26C0  6A05 3E17 7817 BA1B 9BFA
sub   rsa2048/0x4FE01854A428189E 2015-10-31 [A]
      Key fingerprint = E70A D210 1083 51FB 03F5  D4FB 4FE0 1854 A428 189E
```

2.5.1. Main website: https://f-droid.org .

2.5.2. See https://web.archive.org/web/20220405054709/https://f-droid.org/en/about/ .

2.5.3. See https://web.archive.org/web/20171220224805/https://f-droid.org/en/docs/Release_Channels_and_Signing_Keys/ .

2.5.4. See https://web.archive.org/web/20200513031502/https://f-droid.org/F-Droid.apk.asc and https://web.archive.org/web/20171017002616/https://f-droid.org/ .

2.5.3.3 ANDROID client APK signing key (2010–)

A public key used to sign the APK file used to install the ANDROID client.[2.5.5]

```
Owner: CN=Ciaran Gultnieks, OU=Unknown, O=Unknown, L=Wetherby, ST=Unknown, C=UK
Issuer: CN=Ciaran Gultnieks, OU=Unknown, O=Unknown, L=Wetherby, ST=Unknown, C=UK
Serial number: 4c49cd00
Valid from: Fri Jul 23 13:10:24 EDT 2010 until: Tue Dec 08 12:10:24 EST 2037
Certificate fingerprints:

  MD5:  17:C5:5C:62:80:56:E1:93:E9:56:44:E9:89:79:27:86
  SHA1: 05:F2:E6:59:28:08:89:81:B3:17:FC:9A:6D:BF:E0:4B:0F:A1:3B:4E
  SHA256: 43:23:8D:51:2C:1E:5E:B2:D6:56:9F:4A:3A:FB:F5:52:34:18:B8:2E:0A:3E:D1:55:27:70:AB:B9:A9:C9:CC:AB
```

2.5.5. See https://f-droid.org/en/docs/Release_Channels_and_Signing_Keys/ .

2.6 FREEDOMBOX

Last updated 2022-05-14 by STEVEN BALTAKATEI SANDOVAL.

2.6.1 Background

FREEDOMBOX[2.6.1] is an operating system based on GNU/LINUX DEBIAN designed to provide individuals locally-hosted internet services in lieu of popular cloud services such as those offered by GOOGLE, MICROSOFT, FACEBOOK, and APPLE. Services include:

- Publishing and Blogging (via WORDPRESS, IKIWIKI, and MEDIAWIKI).
- File sharing (via BEPASTY and other apps).
- Version control (via GITWEB).
- End-to-end encrypted chat (via MATRIX)

The operating system is designed to run on low power hardware such as single-board computers including the RASPBERRY PI.

The software is maintained by the FREEDOMBOX FOUNDATION which was founded in 2010 by EBEN MOGLEN.[1]

2.6.2 History

2010-08-08. First I.A. snapshot of http://wiki.debian.org/FreedomBox .[2.6.2]

2011-02-17. First I.A. snapshot of http://freedomboxfoundation.org .[2.6.3]

2011-02-15. NEW YORK TIMES publishes article about creation of the FREEDOM BOX FOUNDATION by EBEN MOGLEN.[1]

2011-11-12. Creation date of signing key 36C3 6144 0C9B C971 (SUNIL MOHAN ADAPA).

2015-06-07. Creation date of signing key 77C0 C75E 7B65 0808 (JAMES VALLEROY).

2018-06-06. Creation date of signing key 5D41 53D6 FE18 8FC8 (FREEDOMBOX C.I. server).

2.6.3 Public Key Details

2.6.3.1 Signing key (2015–2019) (36C3 6144 0C9B C971)

Key used by SUNIL MOHAN ADAPA to sign FREEDOMBOX releases between 2015 and 2019.[2.6.4] [2.6.5]

```
pub   rsa4096/0x36C361440C9BC971 2011-11-12 [SC] [expires: 2023-03-23]
      Key fingerprint = BCBE BD57 A11F 70B2 3782  BC57 36C3 6144 0C9B C971
uid                   [ unknown] Sunil Mohan Adapa <0f990da0>
uid                   [ unknown] Sunil Mohan Adapa <040cbdcf>
uid                   [ unknown] Sunil Mohan Adapa <6271c3c3>
sub   rsa4096/0x43EA1CFF0AA7C5F2 2016-06-04 [S] [expires: 2023-03-23]
      Key fingerprint = E713 C363 D672 5A75 AEA5  7481 43EA 1CFF 0AA7 C5F2
sub   rsa4096/0xF9F18B3DA6EF2942 2016-06-04 [A] [expires: 2023-03-23]
      Key fingerprint = DDE7 318C 9541 FB42 E786  8DD0 F9F1 8B3D A6EF 2942
sub   rsa4096/0xF5077A854C1D4B57 2011-11-12 [E] [expires: 2023-03-23]
      Key fingerprint = 83BB 47F9 531E 732C D468  E382 F507 7A85 4C1D 4B57
```

2.6.1. Main website: https://www.freedombox.org/ .

2.6.2. See https://web.archive.org/web/20100808091841/http://wiki.debian.org/FreedomBox .

2.6.3. See https://web.archive.org/web/20110217045826/http://freedomboxfoundation.org/ .

2.6.4. See https://web.archive.org/web/20150926202452/https://wiki.debian.org/FreedomBox/Download .

2.6.5. See https://reboil.com/res/2022/txt/20220514T0328Z..fbx_img_sig_dates.html .

2.6.3.2 Signing key (2016–2017) (77C0 C75E 7B65 0808)

Key used by James Valleroy to sign FREEDOMBOX releases in between 2016 and 2017[2.6.5].

```
pub   rsa4096/0x77C0C75E7B650808 2015-06-07 [SCA] [expires: 2022-11-20]
      Key fingerprint = 7D6A DB75 0F91 0855 8948  4BE6 77C0 C75E 7B65 0808
uid                   [ unknown] James Valleroy <ef8c790c>
uid                   [ unknown] James Valleroy <cc9e6d5c>
uid                   [ unknown] James Valleroy <aed00202>
sub   rsa4096/0x1E3D7658DDA11207 2015-07-03 [A] [expires: 2022-11-20]
      Key fingerprint = 065E 1FA3 78BD C472 41A9  410F 1E3D 7658 DDA1 1207
sub   rsa2048/0x81DD8ABA2A624357 2015-12-22 [A] [expires: 2022-11-20]
      Key fingerprint = 14ED DCE5 A2C1 5C06 E474  0AC0 81DD 8ABA 2A62 4357
sub   rsa4096/0x4972352E25D22BF4 2015-06-07 [E] [expires: 2022-11-20]
      Key fingerprint = 1968 4269 1A27 66F2 1A8B  250C 4972 352E 25D2 2BF4
sub   rsa2048/0xBEE22CB4990F243B 2017-01-14 [A] [expires: 2022-11-20]
      Key fingerprint = BCCD 6658 3F68 0306 8299  79AE BEE2 2CB4 990F 243B
sub   rsa4096/0x1F52746F95690007 2021-08-18 [A] [expires: 2022-08-18]
      Key fingerprint = 34E6 6A4D 0927 9756 A664  609C 1F52 746F 9569 0007
```

2.6.3.3 Signing key (2018–2022) (5D41 53D6 FE18 8FC8)

Key used by the FREEDOMBOX Continuous Integration server to sign FREEDOMBOX releases from 2018 through 2022[2.6.5].

```
pub   rsa4096/0x5D4153D6FE188FC8 2018-06-06 [SC]
      Key fingerprint = 013D 86D8 BA32 EAB4 A669  1BF8 5D41 53D6 FE18 8FC8
uid                   [ unknown] FreedomBox CI (Continuous Integration server) <7c8f3186>
sub   rsa4096/0x24E6B04F3B3AF25D 2018-06-06 [E]
      Key fingerprint = CE3D 01C4 C6D1 FF10 C201  348D 24E6 B04F 3B3A F25D
```

2.7 GITHUB

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.7.1 Background

GITHUB[2.7.1] is a commercial GIT repository hosting service company founded in 2008. It was purchased by MICROSOFT in 2016.[2]

2.7.2 History

2008. GITHUB founded in San Francisco.[2]

2008-03-10. GITHUB parent company LOGICAL AWESOME, LLC registered in San Francisco by Chris Wanstrath.[2.7.2]

2008-05-14. First snapshot of the `https://github.com` website on the INTERNET ARCHIVE.[2.7.3]

2017-08-16. Creation date of the `4AEE 18F8 3AFD EB23` public key according to itself.

2017-11-14. Date of INTERNET ARCHIVE snapshot containing an early link to `https://github.com/web-flow.gpg` from a page on the `help.github.com` domain.[2.7.4] Also the date of a post by GITHUB user `jonathancross`[2.7.5] observing that the `4AEE 18F8 3AFD EB23` key appears to be a new feature[2.7.6]:

> Yeah, just experimented and saw the same thing. Strange new "feature" of GitHub it seems.

2018-06-04. First snapshot of the `4AEE 18F8 3AFD EB23` public key `https://github.com/web-flow.gpg` on the INTERNET ARCHIVE.[2.7.7]

2021-05-25. Public key `4AEE 18F8 3AFD EB23` fingerprint explicitly published at GITHUB documentation website.[2.7.8]

2.7.3 Public Key Details

2.7.3.1 Web-flow commit signing (`4AEE 18F8 3AFD EB23`)

As of 2021-07-19, when a user logs into `github.com` and creates a GIT commit through a web browser, GITHUB will automatically sign the commit against a GPG key[2.7.9] with the fingerprint:

```
pub   rsa2048/0x4AEE18F83AFDEB23 2017-08-16 [SC]
      Key fingerprint = 5DE3 E050 9C47 EA3C F04A  42D3 4AEE 18F8 3AFD EB23
uid                   [ unknown] GitHub (web-flow commit signing) <3ca48f8d>
```

This key is available for download at GITHUB's documentation website at `https://github.com/web-flow.gpg` .[2.7.10] This particular link as well as the full key fingerprint was added to the GITHUB documentation repository in a commit dated 2021-05-25[2.7.11].

2.7.1. Main website: `https://github.com/` .

2.7.2. See `https://businesssearch.sos.ca.gov/Document/RetrievePDF?Id=200807010145-721605` and `https://businesssearch.sos.ca.gov/Document/RetrievePDF?Id=200807010145-2544282` from `https://opencorporates.com/companies/us_ca/200807010145` .

2.7.3. See `https://web.archive.org/web/20080514210148/http://github.com/` .

2.7.4. See `https://web.archive.org/web/20171114055613/https://help.github.com/articles/about-gpg/` .

2.7.5. Key fingerprint `C0C0 7613 2FFA 7695`. Key at `https://github.com/jonathancross.gpg` .

2.7.6. `https://github.com/keepassxreboot/keepassxc/issues/1183#issuecomment-344386172` .

2.7.7. `https://web.archive.org/web/20180604123146/https://github.com/web-flow.gpg` .

2.7.8. See See `https://github.com/github/docs/commit/c4e1cb7a97704f0d90c0d6ed7e52d72b1e4946c1` .

2.7.9. See `https://reboil.com/res/2021/txt/20210719_4AEE18F83AFDEB23..github.asc` or `https://github.com/web-flow.gpg` .

2.7.10. See `https://docs.github.com/en/github/authenticating-to-github/managing-commit-signature-verification/about-commit-signature-verification` .

2.7.11. See `https://github.com/github/docs/commit/c4e1cb7a97704f0d90c0d6ed7e52d72b1e4946c1` .

2.8 GNUPG

Last updated 2022-04-13 by STEVEN BALTAKATEI SANDOVAL.

2.8.1 Background

GNUPG[2.8.1] is a set of privacy-enhancing programs licensed[2.8.2] under the GNU GENERAL PUBLIC LICENSE[2.8.3] designed to encrypt and digitally sign data using the OPENPGP standard[2.8.4]. The name "GNUPG", or "GPG" as its main program gpg is called, is a play on words alluding to the acronym "PGP". PGP, which stands for "Pretty Good Privacy" was a commercial program created by PHIL ZIMMERMANN in 1991 from which the OPENPGP standard was derived.

Compiled binary releases of GNUPG themselves use GNUPG to verify their own integrity. A list of public keys used to sign releases is provided in Table 2.8.1. If a trusted instance of GNUPG is not available, checksums of releases are published on the `https://gnupg.org` website.[2.8.5] That said, GNUPG is often installed by default on GNU/LINUX operating systems, such as DEBIAN (see 2.3), in which software package managers, such as APT, automatically use GNUPG to verify the integrity of downloaded software[2.8.6]. In particular, DEBIAN also verifies developer identities using OPENPGP public keys[2.8.7].

As of 2022, the GNUPG project is maintained by WERNER KOCH (`5288 97B8 2640 3ADA`) and funded by commercial support contracts for a version of GPG4WIN called "GnuPG VS-Desktop".[2.8.8] Prior to 2022, a significant fraction of project funding originated from donations by individuals.[2.8.9]

2.8.2 History

1998-07-07. WERNER KOCH creates first release signing key `68B7 AB89 5754 8DCD`.

1999-01-29. Date of early webpage I.A. snapshot at `http://www.k.shuttle.de` domain.[2.8.10]

1999-09-07. GNUPG version `1.0.0` released.[2.8.11]

2000-03-03. Date of early I.A. snapshot of `http://www.gnupg.org` domain.[2.8.12]

2001-05-03. Date of early I.A. snapshot of `https://www.g10.code.com` domain.[2.8.13]

2006-11-11. GNUPG version `2.0.0` released.[2.8.14]

2.8.1. Main website: `https://gnupg.org` .

2.8.2. See `https://web.archive.org/web/20070708182544/http:/lists.gnupg.org/pipermail/gnupg-announce/2007q3/000255.html` .

2.8.3. See `https://www.gnu.org/licenses/gpl-3.0.en.html` .

2.8.4. See RFC4880: `https://www.ietf.org/rfc/rfc4880.txt` .

2.8.5. See `https://gnupg.org/download/integrity_check.html` .

2.8.6. See `https://wiki.debian.org/SecureApt` .

2.8.7. See `https://www.debian.org/devel/join/nm-step2` .

2.8.8. See `https://gnupg.org/blog/20220102-a-new-future-for-gnupg.html` .

2.8.9. See `https://gnupg.org/donate/kudos.html` .

2.8.10. See `https://web.archive.org/web/20000303105255/http://www.gnupg.org/` .

2.8.11. See `https://web.archive.org/web/20040318173823/http://lists.gnupg.org/pipermail/gnupg-announce/1999q3/000037.html` .

2.8.12. See `https://web.archive.org/web/20000303105255/http://www.gnupg.org/` .

2.8.13. See `https://web.archive.org/web/20010503130044/http://www.g10code.com/` .

2.8.14. See `https://web.archive.org/web/20061117172350/http://lists.gnupg.org/pipermail/gnupg-announce/2006q4/000239.html` .

2.8.3 Public Key Details

Various public keys have been used to sign compiled binary releases of GNUPG. Below are the keys valid as of 2022.

2.8.3.1 Release signing key - WERNER KOCH (2020–) (5288 97B8 2640 3ADA)

Key used by WERNER KOCH to sign releases of GNUPG since 2020.

```
pub   ed25519/0x528897B826403ADA 2020-08-24 [SC] [expires: 2030-06-30]
      Key fingerprint = 6DAA 6E64 A76D 2840 571B  4902 5288 97B8 2640 3ADA
uid                   [ unknown] Werner Koch (dist signing 2020)
```

2.8.3.2 Release signing key - NIIBE YUTAKA (2021–) (E98E 9B2D 19C6 C8BD)

Key used by NIIBE YUTAKA to sign releases of GNUPG since 2021.

```
pub   ed25519/0xE98E9B2D19C6C8BD 2021-05-19 [SC] [expires: 2027-04-04]
      Key fingerprint = AC8E 115B F73E 2D8D 47FA  9908 E98E 9B2D 19C6 C8BD
uid                   [ unknown] Niibe Yutaka (GnuPG Release Key)
```

2.8.3.3 Release signing key - ANDRE HEINECKE (2017–) (BCEF 7E29 4B09 2E28)

Key used by ANDRE HEINECKE to sign releases of GNUPG since 2017.

```
pub   rsa3072/0xBCEF7E294B092E28 2017-03-17 [SC] [expires: 2027-03-15]
      Key fingerprint = 5B80 C575 4298 F0CB 55D8  ED6A BCEF 7E29 4B09 2E28
uid                   [ unknown] Andre Heinecke (Release Signing Key)
```

2.8.3.4 Release signing key - GNUPG.COM (2021–) (549E 695E 905B A208)

Key used by the G10 CODE GMBH organization under the GNUPG.COM brand to sign releases of GNUPG since 2021.[2.8.23]

```
pub   brainpoolP256r1/0x549E695E905BA208 2021-10-15 [SC] [expires: 2029-12-31]
      Key fingerprint = 02F3 8DFF 731F F97C B039  A1DA 549E 695E 905B A208
uid                   [ unknown] GnuPG.com (Release Signing Key 2021)
sub   brainpoolP256r1/0x9CDA5DC48371F0E3 2021-10-15 [A] [expires: 2029-12-31]
      Key fingerprint = 6819 7595 44AC 985D 8D52  6066 9CDA 5DC4 8371 F0E3
```

Cr. Date	Long ID	UID	Used	Link
1998-07-07	68B7 AB89 5754 8DCD	Werner Koch (gnupg sig)	1998–2005 [2.8.15]	[2.8.16]
2006-01-01	53B6 20D0 1CE0 C630	Werner Koch (dist sig)	1996–2010 [2.8.15]	[2.8.17]
2011-01-12	249B 39D2 4F25 E3B6	Werner Koch (dist sig)	2011–2021	[2.8.17]
2014-10-29	0437 6F3E E085 6959	David Shaw (GnuPG Release Signing Key)	2015–2020	[2.8.18]
2014-10-29	2071 B08A 33BD 3F06	NIIBE Yutaka (GnuPG Release Key)	2015–2021	[2.8.18]
2014-10-19	8A86 1B1C 7EFD 60D9	Werner Koch (Release Signing Key)	2015–2017	[2.8.18]
2017-03-17	BCEF 7E29 4B09 2E28	Andre Heinecke (Release Signing Key)	2017–	[2.8.19]
2020-08-24	5288 97B8 2640 3ADA	Werner Koch (dist signing 2020)	2020–	[2.8.20]
2021-05-19	E98E 9B2D 19C6 C8BD	Niibe Yutaka (GnuPG Release Key)	2021–	[2.8.21]
2021-10-15	549E 695E 905B A208	GnuPG.com (Release Signing Key 2021)	2021–	[2.8.22]

Table 2.8.1. A list of keys used to sign GNUPG releases. Keys identified from INTERNET ARCHIVE snapshots of http://www.gnupg.org/signature_key.html .

2.8.15. Date span source: https://web.archive.org/web/20131123175952/http://www.gnupg.org:80/signature_key.html .
2.8.16. See https://web.archive.org/web/20041113170551/http://www.gnupg.org/signature_key.html .
2.8.17. See https://web.archive.org/web/20131123175952/http://www.gnupg.org:80/signature_key.html .
2.8.18. See https://web.archive.org/web/20150503220844/https://www.gnupg.org/signature_key.html .
2.8.19. See https://web.archive.org/web/20180515231121/https://gnupg.org/signature_key.html .
2.8.20. See https://web.archive.org/web/20200917215036/https://gnupg.org/signature_key.html .
2.8.21. See https://web.archive.org/web/20210923054234/https://www.gnupg.org/signature_key.html .
2.8.22. See https://web.archive.org/web/20211018075758/https://gnupg.org/signature_key.html .

2.8.23. See https://gnupg.org/blog/20220102-a-new-future-for-gnupg.html for origin of GNUPG.COM brand.

2.9 QUBES OS

Last updated 2022-04-11 by STEVEN BALTAKATEI SANDOVAL.

2.9.1 Background

QUBES OS[2.9.1] is a privacy-focused operating system made by INVISIBLE THINGS LABS. Privacy is enhanced by isolating programs so each runs in its own virtual machine environment. The project uses OPENPGP to sign release files[2.9.2].

The project was founded in 2010 by JOANNA RUTKOWSKA (5FA6 C3E4 D9AF BB99). As of 2022, the project lead is MAREK MARCZYKOWSKI-GÓRECKI (DB8F D31C CAD7 D72C).

2.9.2 History

2010-04-01. Creation date of the Qubes Master Signing Key (DDFA 1A3E 3687 9494; a.k.a. QMSK).

2010-04-09. First snapshot on IA of https://qubes-os.org [2.9.3].

2010-04-12. Early publication on IA of the QMSK (DDFA 1A3E 3687 9494) full fingerprint.[2.9.4]

2012-03-31. Creation date of the Release 1 signing key (EA01 201B 2110 93A7).

2012-11-15. Creation date of the Release 2 signing key (0C73 B9D4 0A40 E458).

2014-11-19. Creation date of the Release 3 signing key (CB11 CA1D 03FA 5082).

2017-03-06. Creation date of the Release 4 signing key (1848 792F 9E27 95E9).

2.9.3 Public Key Details

2.9.3.1 Qubes Master Signing Key (DDFA 1A3E 3687 9494)

Key used by the Qubes OS Project to sign keys of official team members and release signing keys. A procedure for downloading and verifying this key is available on the QUBES OS website.[2.9.5]

```
pub   rsa4096/0xDDFA1A3E36879494 2010-04-01 [SC]
      Key fingerprint = 427F 11FD 0FAA 4B08 0123  F01C DDFA 1A3E 3687 9494
uid                   [ unknown] Qubes Master Signing Key
```

2.9.3.2 Release 1 Signing Key (EA01 201B 2110 93A7)

Key used to sign Release 1 of QUBES OS[2.9.6].

```
pub   rsa4096/0xEA01201B211093A7 2012-03-31 [SC]
      Key fingerprint = FFED 4FD8 E49E 79F3 9C83  FD81 EA01 201B 2110 93A7
uid                   [ unknown] Qubes OS Release 1 Signing Key
```

2.9.3.3 Release 2 Signing Key (0C73 B9D4 0A40 E458)

Key used to sign Release 2 of QUBES OS[2.9.7].

```
pub   rsa4096/0x0C73B9D40A40E458 2012-11-15 [SC]
      Key fingerprint = 3F01 DEF4 9719 158E F862  66F8 0C73 B9D4 0A40 E458
uid                   [ unknown] Qubes OS Release 2 Signing Key
```

2.9.1. Main website: https://www.qubes-os.org/ .

2.9.2. See https://www.qubes-os.org/downloads/ .

2.9.3. See https://web.archive.org/web/20100409054657/http://qubes-os.org/Home.html .

2.9.4. See https://web.archive.org/web/20100412080416/http://www.qubes-os.org/trac/wiki/VerifyingSignatures .

2.9.5. See https://www.qubes-os.org/security/verifying-signatures/ .

2.9.6. See https://blog.invisiblethings.org/2012/09/03/introducing-qubes-10.html .

2.9.7. See https://blog.invisiblethings.org/2014/09/26/announcing-qubes-os-release-2.html .

2.9.3.4 Release 3 Signing Key (CB11 CA1D 03FA 5082)

Key used to sign Release 3 of QUBES OS[2.9.8].

```
pub   rsa4096/0xCB11CA1D03FA5082 2014-11-19 [SC]
      Key fingerprint = C522 61BE 0A82 3221 D94C  A1D1 CB11 CA1D 03FA 5082
uid                   [ unknown] Qubes OS Release 3 Signing Key
```

2.9.3.5 Release 4 Signing Key (1848 792F 9E27 95E9)

Key used to sign Release 4 of QUBES OS[2.9.9].

```
pub   rsa4096/0x1848792F9E2795E9 2017-03-06 [SC]
      Key fingerprint = 5817 A43B 283D E5A9 181A  522E 1848 792F 9E27 95E9
uid                   [ unknown] Qubes OS Release 4 Signing Key
```

2.9.8. See https://www.qubes-os.org/doc/releases/3.0/release-notes/ .

2.9.9. See https://www.qubes-os.org/doc/releases/4.0/release-notes/ .

2.10 RASPIBLITZ

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.10.1 Background

RASPIBLITZ[2.10.1] is a software package designed to facilitate operation of a LIGHTNING NETWORK and BITCOIN node. The software is version controlled using GIT, with the main git repository available at GITHUB.[2.10.2] As of 2021-07-18, the principal maintainer appears to be CHRISTIAN "ROOTZOL" ROTZOLL[2.10.3].

2.10.2 History

2019-09-03. The creation date of `rootzol`'s `1C73 060C 7C17 6461` public key.

2019-09-05. ROOTZOL added their public key fingerprint `1C73 060C 7C17 6461` to the FAQ of the RASPIBLITZ GITHUB repository.[2.10.4] They linked their `keybase.io` page as a source of the public key.

2020-10-31. The first snapshot of the `raspiblitz.org` website appeared on the Internet Archive.[2.10.5]

2021-02-07. Andreas Antonopoulos posted a YouTube video identifying RASPIBLITZ as a popular Bitcoin full node software package.[2.10.6]

2021-05-18. ROOTZOL added their public key fingerprint `1C73 060C 7C17 6461` to the `README` of the RASPIBLITZ GITHUB repository.

2.10.3 Public Key Details

2.10.3.1 CHRISTIAN "ROOTZOL" ROTZOLL (`1C73 060C 7C17 6461`)

ROOTZOL's PGP key[2.10.7] may be downloaded from their Keybase page.[2.10.8]. Their fingerprint information is as follows:

```
pub   rsa4096/0x1C73060C7C176461 2019-09-03 [C]
      Key fingerprint = 92A7 46AE 33A3 C186 D014  BF5C 1C73 060C 7C17 6461
uid                   [ unknown] Christian Rotzoll <bca61f43>
sub   rsa4096/0xAA9DD1B5CC5647DA 2019-09-03 [S] [expires: 2022-10-26]
      Key fingerprint = C0EE 6145 31A4 16B4 CDB7  A2D7 AA9D D1B5 CC56 47DA
sub   rsa4096/0xD40D94E6C7C9B4D9 2019-09-03 [E] [expires: 2022-10-26]
      Key fingerprint = A7D4 DA62 95EC 365E 4CCF  05FC D40D 94E6 C7C9 B4D9
sub   rsa4096/0x1C29DC2F8D764F9A 2019-09-03 [A] [expires: 2022-10-26]
      Key fingerprint = AA80 506C FCAB 3405 84C4  9F0E 1C29 DC2F 8D76 4F9A
```

2.10.1. Main website: https://raspiblitz.org/ .

2.10.2. See https://github.com/rootzoll/raspiblitz .

2.10.3. Their public key 0x1C73060C7C176461 is available at: https://keybase.io/rootzoll .

2.10.4. See https://github.com/rootzoll/raspiblitz/commit/75ebdd8d571cccc427b5d023a25c6e2e9e8a2da2 .

2.10.5. See https://web.archive.org/web/20201031223643/https://raspiblitz.org/ .

2.10.6. See https://www.youtube.com/watch?v=AXUfwvhr3lg&t=26m27s .

2.10.7. See https://reboil.com/res/2021/txt/20210719_0x1C73060C7C176461..raspiblitz_rootzol.asc

2.10.8. See https://keybase.io/rootzoll/pgp_keys.asc .

2.11 SATOSHI LABS

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.11.1 Background

SATOSHI LABS[2.11.1] is a company that produces cryptocurrency hardware wallets called TREZOR[2.11.2]. These devices enable a user to privately manage their private keys necessary to create transactions. Publishing transactions and viewing current balances typically requires software running on a computer connected to the internet. SATOSHI LABS uses an OpenPGP key to sign these software packages published on their website https://trezor.io .

SATOSHI LABS was founded in 2013 by MAREK "SLUSH" PALATINUS, PAVOL "STICK" RUSNÁK, and ALENA VRANOVA.[2.11.3] It is based in Prague, Czech Republic.

As of 2022-01-03, the primary TREZOR program requiring verification is TREZOR SUITE.

2.11.2 History

2012-03-07. Creation date of PAVOL RUSNÁK's personal PGP key (91F3 B339 B9A0 2A3D).

2014-07-18. First snapshot of https://mytrezor.com appears on the INTERNET ARCHIVE.[2.11.4]

2017-01-11. mytrezor.com , buytrezor.com , and other domains migrated to https://trezor.io .[2.11.5]

2017-01-28. The first snapshot of https://trezor.io appears on the INTERNET ARCHIVE.[2.11.6]

2020-10-20. Creation date of the 2020 signing key (26A3 A566 62F0 E7E2).

2021-01-04. Creation date of the 2021 signing key (E21B 6950 A2EC B65C).

2021-07-14. TREZOR SUITE launched[2.11.7] in order to replace an older web wallet implementation.[2.11.8]

2.11.3 Public Key Details

2.11.3.1 PAVOL RUSNÁK (91F3 B339 B9A0 2A3D)

A key[2.11.9] used by a developer named PAVOL "STICK" RUSNÁK.[2.11.10] This key has been used to sign TREZOR software in the past[2.11.11] such as TREZOR BRIDGE[2.11.12] and other various GITHUB commits.

```
pub   rsa4096/0x91F3B339B9A02A3D 2012-03-07 [SC] [expires: 2024-06-16]
      Key fingerprint = 86E6 792F C27B FD47 8860  C110 91F3 B339 B9A0 2A3D
uid                   [ unknown] Pavol Rusnák <343a72bf>
uid                   [ unknown] Pavol Rusnák <707f7617>
uid                   [ unknown] Pavol Rusnák <5144f42a>
uid                   [ unknown] Pavol Rusnák <5aef3feb>
uid                   [ unknown] [jpeg image of size 2449]
sub   rsa4096/0x22AF226D38DC1F4D 2012-03-07 [E] [expires: 2024-06-16]
      Key fingerprint = E177 6F65 0601 E596 9E7F  9E25 22AF 226D 38DC 1F4D
```

2.11.1. Main website: https://satoshilabs.com/ .

2.11.2. Trezor website: https://trezor.io/ .

2.11.3. See https://web.archive.org/web/20140627154535/http://satoshilabs.com/team/ .

2.11.4. See https://web.archive.org/web/20140718104157/https://mytrezor.com/ .

2.11.5. See https://web.archive.org/web/20201111170337/https://blog.trezor.io/new-trezor-io-55cf687c88d5?gi=3481ee5b4637 .

2.11.6. See https://web.archive.org/web/20170128023418/https://trezor.io/ .

2.11.7. See https://blog.trezor.io/trezor-suite-launches-8958c1d37d33 .

2.11.8. See https://github.com/trezor-graveyard .

2.11.9. Download key at https://rusnak.io/public/pgp.txt .

2.11.10. Twitter: https://twitter.com/pavolrusnak .

2.11.11. See https://github.com/trezor/trezord-go/issues/211 .

2.11.12. See https://github.com/trezor/webwallet-data/tree/master/bridge .

2.11.3.2 2020 Signing Key (26A3 A566 62F0 E7E2)

A key[2.11.13] used to sign the software required by a PC to communicate with the TREZOR product line. Expired as of 2021-01-01.

```
pub   rsa4096/0x26A3A56662F0E7E2 2020-10-20 [SC] [expired: 2021-01-01]
      Key fingerprint = 5406 7D8B BF00 5541 81B5  AB8F 26A3 A566 62F0 E7E2
uid                   [ expired] SatoshiLabs 2020 Signing Key
```

2.11.3.3 2021 Signing Key (E21B 6950 A2EC B65C)

A key[2.11.14] used to sign the software required by a PC to communicate with the Trezor product line.

```
pub   rsa4096/0xE21B6950A2ECB65C 2021-01-04 [SC]
      Key fingerprint = EB48 3B26 B078 A4AA 1B6F  425E E21B 6950 A2EC B65C
uid                   [ unknown] SatoshiLabs 2021 Signing Key
```

2.11.13. Download key at https://trezor.io/security/satoshilabs-2020-signing-key.asc .

2.11.14. Download key at https://trezor.io/security/satoshilabs-2021-signing-key.asc .

2.12 TAILS

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.12.1 Background

TAILS[2.12.1] is a DEBIAN-based GNU/LINUX operating system designed to preserve user anonymity through default use of TOR for all network traffic.

TAILS is an acronym for **T**HE (**A**MNESIC) **I**NCOGNITO **L**IVE **S**YSTEM. The name reflects the fact that the project is the result of a 2010 merger between two TOR-related projects known as AMNESIA[2.12.2] and INCOGNITO[2.12.3]. In addition to differing sets of developers, the two projects were based on two different GNU/LINUX operating systems: AMNESIA used DEBIAN and INCOGNITO used GENTOO.

2.12.2 History

2009-06-20. Date of first commit in AMNESIA repository.[2.12.4]

2010-04-07. INCOGNITO[2.12.5] and AMNESIA merge.[2.12.6] Renamed "THE (AMNESIC) INCOGNITO LIVE SYSTEM".[2.12.7]

2010-07-16. First I.A. snapshot of `http://amnesia.boum.org` .[2.12.8]

2010-10-07. Creation date of the signing key `1202 821C BE2C D9C1`.

2011-03-17. First I.A. snapshot of `https://tails.boum.org` .[2.12.9]

2015-01-18. Creation date of the signing key `DBB8 02B2 58AC D84F`.

2015-03-16. Signing key `1202 821C BE2C D9C1` officially retired and replaced by `DBB8 02B2 58AC D84F`.[2.12.10]

2.12.3 Public Key Details

As of 2022, several public keys are associated with the TAILS project.[2.12.11] This section describes only the current signing key (See 2.12.3.1), mailing list encryption key (See 2.12.3.3), and the 2010–2015 signing key (See 2.12.3.2).

2.12.3.1 Signing key (2015–) (`DBB8 02B2 58AC D84F`)

Key used by the TAILS developers to sign released `.iso` installation images since TAILS version *v1.3.1* which was released in 2015.[2.12.12] The public key may be downloaded here[2.12.13].

2.12.1. Main website: `https://tails.boum.org/` .

2.12.2. See `https://web.archive.org/web/20100716170307/http://amnesia.boum.org/` .

2.12.3. See phobos. "Incognito and The Tor Project sign a licensing agreement". Website: `blog.torproject.org` . Date: 2008-06-27. Archive URL: `https://web.archive.org/web/20081120073057/http://blog.torproject.org/blog/incognito-and-tor-project-sign-licensing-agreement` . Archive date: 2008-11-20.

2.12.4. See `https://gitlab.tails.boum.org/tails/tails/-/blob/345a927fbd6aa18a2bcd13331cbc2e22ef2e0639/config/chroot_local-includes/usr/share/doc/amnesia/Changelog` .

2.12.5. See `https://web.archive.org/web/20100108093152/http://anonymityanywhere.com/` .

2.12.6. See: anonym. "Incognito + Amnesia = The (Amnesic) Incognito Live System". Date: 2010-04-07. Archive date: 2010-07-28. Archive URL: `https://web.archive.org/web/20100728224716/http://www.anonymityanywhere.com:80/incognito` .

2.12.7. See "new project name". Website: `amnesia.boum.org` . Date: 2010-04-07. Archive date: 2010-08-17. Archive URL: `https://web.archive.org/web/20100817180857/http://amnesia.boum.org/news/new_project_name/` .

2.12.8. See `https://web.archive.org/web/20100716170307/http://amnesia.boum.org/` .

2.12.9. See `https://web.archive.org/web/20110317013911/https://tails.boum.org/` .

2.12.10. See `https://web.archive.org/web/20150316172733/https://tails.boum.org/news/signing_key_transition/index.en.html` .

2.12.11. See "OpenPGP keys". Date accessed: 2022-03-12. `https://tails.boum.org/doc/about/openpgp_keys/index.en.html` .

2.12.12. See "Tails 1.3.1 is out". Website: `tails.boum.org` . Date: 2015-03-23. Archive URL: `https://web.archive.org/web/20150402101752/https://tails.boum.org/news/version_1.3.1/index.en.html` . Archive date: 2015-04-02.

2.12.13. See `https://tails.boum.org/tails-signing.key` .

```
pub   rsa4096/0xDBB802B258ACD84F 2015-01-18 [C] [expires: 2023-01-07]
      Key fingerprint = A490 D0F4 D311 A415 3E2B  B7CA DBB8 02B2 58AC D84F
uid                   [ unknown] Tails developers (offline long-term identity key) <16f58847>
uid                   [ unknown] Tails developers <16f58847>
sub   rsa4096/0xD21DAD38AF281C0B 2017-08-28 [S] [expires: 2023-01-07]
      Key fingerprint = 0546 9FB8 5EAD 6589 B43D  41D3 D21D AD38 AF28 1C0B
sub   ed25519/0x90B2B4BD7AED235F 2017-08-28 [S] [expires: 2023-01-07]
      Key fingerprint = CD4D 4351 AFA6 933F 574A  9AFB 90B2 B4BD 7AED 235F
sub   rsa4096/0x7BFBD2B902EE13D0 2021-10-14 [S] [expires: 2023-01-07]
      Key fingerprint = 753F 9013 77A3 09F2 731F  A33F 7BFB D2B9 02EE 13D0
```

2.12.3.2 Signing key (2010–2015) (1202 821C BE2C D9C1)

Key used by the TAILS developers to sign released images starting with TAILS *v0.6*[2.12.14] until and including *v1.3*. The public key may be downloaded here[2.12.15]. This key was retired from use in 2015 and replaced with key DBB8 02B2 58AC D84F.[2.12.16]

```
pub   rsa4096/0x1202821CBE2CD9C1 2010-10-07 [SC] [expired: 2015-04-30]
      Key fingerprint = 0D24 B36A A9A2 A651 7878  7645 1202 821C BE2C D9C1
uid                   [ expired] Tails developers (signing key) <16f58847>
```

2.12.3.3 Mailing list key (2009–) (1D29 75ED F93E 735F)

Key recommended by the TAILS developers to be used to encrypt emails sent to their encrypted mailing list since at least 2011[2.12.17], possibly 2009[2.12.18], assuming the main repository timetamps are trustworthy. Until TAILS *v0.5* and *v0.6~rc3*, released images were signed using this key.[2.12.19] The public key may be downloaded here[2.12.20].

```
pub   rsa4096/0x1D2975EDF93E735F 2009-08-14 [SC] [expires: 2023-03-03]
      Key fingerprint = 09F6 BC8F EEC9 D8EE 005D  BAA4 1D29 75ED F93E 735F
uid                   [ unknown] Tails developers (Schleuder mailing-list) <16f58847>
uid                   [ unknown] Tails list (schleuder list) <a6d8622b>
uid                   [ unknown] Tails list (schleuder list) <87eb09d2>
sub   rsa4096/0xD843C2F5E89382EB 2009-08-14 [E] [expires: 2023-03-03]
      Key fingerprint = C394 8FE7 B604 C611 4E29  4DDF D843 C2F5 E893 82EB
```

2.12.14. See https://web.archive.org/web/20111205083704/https://tails.boum.org/doc/about/openpgp_keys/index.en.html .

2.12.15. See https://web.archive.org/web/20141006010041/https://tails.boum.org/tails-signing.key .

2.12.16. See "Transition to a new OpenPGP signing key". Website: tails.boum.org . Date: 2015-03-16. Archive URL: https://web.archive.org/web/20150316172733/https://tails.boum.org/news/signing_key_transition/index.en.html . Archive date: 2015-03-16.

2.12.17. See https://web.archive.org/web/20110318070814/http://tails.boum.org/GnuPG_key/index.en.html .

2.12.18. See https://gitlab.tails.boum.org/tails/tails/-/blob/195b39cbf409fa7a8763cc6a6c5f91386db6735b/debian/changelog .

2.12.19. See https://web.archive.org/web/20111205083704/https://tails.boum.org/doc/about/openpgp_keys/index.en.html .

2.12.20. See https://tails.boum.org/tails-email.key .

2.13 TOR BROWSER

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.13.1 Background

TOR BROWSER[2.13.1] is a browser software package that permits visiting websites with anonymity effected by onion routing. Although various[2.13.2] PGP keys have been used to sign various releases and archives, the 4E2C 6E87 9329 8290 key has been used for the main TOR BROWSER installer since at least 2015.

2.13.2 History

2008-01-30. STEVEN J. MURDOCH announces development of TOR BROWSER.[2.13.3]

2014-12-15. Creation date of the 4E2C 6E87 9329 8290 binary signing key.

2019-06-29. Copies of the main release signing key 4E2C 6E87 9329 8290 maintained by various keyservers suffered a certificate spamming attack.[2.13.4] Other high-profile PGP keys were also affected at this time.[2.13.5]

2.13.3 Public Key Details

2.13.3.1 Release Signing Key (2015–) (4E2C 6E87 9329 8290)

Public key used for signing TOR BROWSER releases since at least 2015-03-15[2.13.6] until 2022-03-06[2.13.7].

```
pub   rsa4096/0x4E2C6E8793298290 2014-12-15 [C] [expires: 2025-07-21]
      Key fingerprint = EF6E 286D DA85 EA2A 4BA7  DE68 4E2C 6E87 9329 8290
uid                   [ unknown] Tor Browser Developers (signing key) <85e84cd9>
sub   rsa4096/0xE53D989A9E2D47BF 2021-09-17 [S] [expires: 2023-09-17]
      Key fingerprint = 6131 88FC 5BE2 176E 3ED5  4901 E53D 989A 9E2D 47BF
```

2.13.1. Main website: https://www.torproject.org .

2.13.2. See https://web.archive.org/web/20210713130216/https://2019.www.torproject.org/docs/signing-keys.html.en .

2.13.3. See https://lists.torproject.org/pipermail/tor-talk/2008-January/007837.html .

2.13.4. See https://nvd.nist.gov/vuln/detail/CVE-2019-13050 .

2.13.5. See https://gist.github.com/rjhansen/67ab921ffb4084c865b3618d6955275f#gistcomment-2959168 .

2.13.6. See https://web.archive.org/web/20150315013830/https://www.torproject.org/docs/verifying-signatures.html.en .

2.13.7. See https://web.archive.org/web/20220221121737/https://support.torproject.org/tbb/how-to-verify-signature/ .

2.14 VERACRYPT

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.14.1 Background

VERACRYPT[2.14.1] is an encryption software package compatible with WINDOWS, MACOS, and GNU/LINUX operating systems. The program is primarily maintained by MOUNIR IDRASSI.

VERACRYPT is a fork of TRUECRYPT made in 2013.

2.14.2 History

2013-06-29. First I.A. snapshot of `http://veracrypt.codeplex.com`, VERACRYPT's first public repository address.

2014-07-15. Early mention of full `EB55 9C7C 54DD D393` fingerprint on `www.idrix.fr` website.[2.14.2]

2014-05-28. TRUECRYPT development halt announcement posted on SOURCEFORGE repository.[2.14.3]

2014-06-27. Creation date of signing key `EB55 9C7C 54DD D393`.

2016-10-17. VERACRYPT *v1.18* audited by QUARKSLAB and *v1.19* released to fix most reported vulnerabilities.[2.14.4].

2017-05-25. First I.A. snapshot of `https://veracrypt.fr` [2.14.5], the new website made in response to MICROSOFT shutting down `codeplex.com` in 2017[2.14.6].

2018-09-11. Creation date of signing key `821A CD02 680D 16DE`.

2018-09-12. Signing key `EB55 9C7C 54DD D393` retired and replaced by key `821A CD02 680D 16DE` via a transition statement signed by both keys.[2.14.7]

2.14.3 Public Key Details

2.14.3.1 Signing key (2018–) (`821A CD02 680D 16DE`)

Key used to sign VERACRYPT releases since version *v1.23* in 2018. A copy of this key can be downloaded here[2.14.8].

```
pub   rsa4096/0x821ACD02680D16DE 2018-09-11 [SC]
      Key fingerprint = 5069 A233 D55A 0EEB 174A  5FC3 821A CD02 680D 16DE
uid                   [ unknown] VeraCrypt Team (2018 - Supersedes Key ID=0x54DDD393) <8042d942>
sub   rsa4096/0x200B5A9D26878A32 2018-09-11 [E]
      Key fingerprint = BB33 5DCA 0D75 325C 6126  BAB6 200B 5A9D 2687 8A32
sub   rsa4096/0x0F5AACD65483D029 2018-09-11 [A]
      Key fingerprint = 6022 69E6 D482 C250 0D1C  2D87 0F5A ACD6 5483 D029
```

2.14.1. Main website: `https://veracrypt.fr` .

2.14.2. See `https://web.archive.org/web/20140715152305/http://www.idrix.fr:80/Root/content/category/7/32/60` .

2.14.3. Goodin, Dan. "'Truecrypt is not secure,' official SourceForge page abruptly warns". Date: 2014-05-28. URL: `https://arstechnica.com/information-technology/2014/05/truecrypt-is-not-secure-official-sourceforge-page-abruptly-warns/` . Access date: 2022-03-12. Archive URL: `https://web.archive.org/web/20140529084822/http://arstechnica.com/security/2014/05/truecrypt-is-not-secure-official-sourceforge-page-abruptly-warns/` . Archive date: 2014-05-29.

2.14.4. ostifadmin. "The Veracrypt Audit Results". Website: `ostif.org` . Date: 2016-10-17. URL: `https://ostif.org/the-veracrypt-audit-results/` . Archive URL: `https://web.archive.org/web/20161017182455/https://ostif.org/the-veracrypt-audit-results/` . Archive date: 2016-10-17.

2.14.5. See `https://web.archive.org/web/20170525235647/https://www.veracrypt.fr/en/Home.html` .

2.14.6. Harry, Brian. "Shutting down CodePlex". Website: `devblogs.microsoft.com` . Date: 2017-03-31. Access date: 2022-03-12. `https://devblogs.microsoft.com/bharry/shutting-down-codeplex/` .

2.14.7. See `https://web.archive.org/web/20181223051800/https://veracrypt.fr/pgp-key-transition-2018-09-12.txt` .

2.14.8. See `https://web.archive.org/web/20220211073947/https://www.idrix.fr/VeraCrypt/VeraCrypt_PGP_public_key.asc` .

2.14.3.2 Signing key (2014–2018) (EB55 9C7C 54DD D393)

Key used to sign VERACRYPT releases prior to version *v1.23* in 2018. This key was used to sign the version of VERACRYPT audited by QUARKSLAB in 2016 (*v1.18*). A 2014 copy of this public key is available here[2.14.9].

```
pub   rsa4096/0xEB559C7C54DDD393 2014-06-27 [SCE]
      Key fingerprint = 993B 7D7E 8E41 3809 828F  0F29 EB55 9C7C 54DD D393
uid                   [ unknown] VeraCrypt Team <8042d942>
```

2.14.9. See https://web.archive.org/web/20200307044514/https://www.idrix.fr/VeraCrypt/VeraCrypt_PGP_public_key_2014.asc .

2.15 YOUTUBE-DL

Last updated 2022-03-12 by STEVEN BALTAKATEI SANDOVAL.

2.15.1 Background

YOUTUBE-DL[2.15.1] is a PYTHON2-based[2.15.2] program that can be used to download audio-visual media files from sites including, but not limited to, YOUTUBE. The software gained notoreity in 2020 when GITHUB took down the project page upon receiving a DMCA takedown notice issued by the RIAA.[2.15.3]

As of 2021, the project maintainer was SERGEY M. (2C39 3E0F 18A9 236D).

Since 2021-12-25, the core developer is REMITA AMINE[2.15.4] (?).

2.15.2 History

2008-07-21. First commit in the main project GIT repository published by RICARDO GARCIA.[2.15.5]

2013-08-01. First image of the homepage https://yt-dl.org appears on the INTERNET ARCHIVE.

2020-10-23. GITHUB project page taken down due to DCMA takedown notice[2.15.6] issued by the RIAA.[2.15.7]

2020-11-16. GITHUB page for YOUTUBE-DL reinstated.[2.15.8]

2021-12-25. The only active developer is REMITA AMINE (?).[2.15.9]

2022-01-29. The project announced[2.15.10] that it is seeking a new maintainer, that YOUTUBE-DL would continue to support PYTHON2, and that the fork YT-DLP created by PUKKANDAN (7EEE 9E1E 817D 0A39) would support PYTHON3.

2.15.3 Public Key Details

2.15.3.1 Binary signing key. SERGEY M. (2C39 3E0F 18A9 236D)

The binary signing key used to sign releases as of 2021. Owned by Sergey M.

```
pub   rsa4096/0x2C393E0F18A9236D 2016-04-09 [SC]
      Key fingerprint = ED7F 5BF4 6B3B BED8 1C87  368E 2C39 3E0F 18A9 236D
uid                   [ unknown] Sergey M. <7345ddad>
sub   rsa4096/0xC3A4FE63297B1CE1 2016-04-09 [E]
      Key fingerprint = 9AA4 FB39 3AF2 73FF 56F9  8251 C3A4 FE63 297B 1CE1
```

2.15.3.2 Binary signing key. PHILIPP HAGEMEISTER (F5EA B582 FAFB 085C)

A binary signing key used by Philipp Hagemeister to sign releases sometime before 2021.[2.15.11]

```
pub   dsa1024/0xF5EAB582FAFB085C 2006-10-23 [SCA] [expired: 2015-12-31]
      Key fingerprint = 0600 E1DB 6FB5 3A5D 95D8  FC0D F5EA B582 FAFB 085C
uid                   [ expired] Philipp Hagemeister <6a728fcb>
uid                   [ expired] Philipp Hagemeister <9482dbb6>
```

2.15.1. Main website: https://yt-dl.org .

2.15.2. See https://developers.slashdot.org/story/22/01/30/003205/youtube-dl-forks-to-continue-supporting-older-versions-of-python .

2.15.3. See https://www.zdnet.com/article/riaa-blitz-takes-down-18-github-projects-used-for-downloading-youtube-videos/ .

2.15.4. See https://github.com/remitamine . Created YT-DLP commit 80d41482 signed by E0DE 62EF 9A9B FAB2.

2.15.5. See https://github.com/ytdl-org/youtube-dl/commit/4fa74b5252a23c2890ddee52b8ee5811b5bb2987 .

2.15.6. See https://github.com/github/dmca/blob/master/2020/10/2020-10-23-RIAA.md .

2.15.7. See https://web.archive.org/web/20201023194520/https://github.com/ytdl-org/youtube-dl .

2.15.8. See https://github.blog/2020-11-16-standing-up-for-developers-youtube-dl-is-back/ .

2.15.9. See https://web.archive.org/web/20211225064545/https://ytdl-org.github.io/youtube-dl/about.html .

2.15.10. See https://github.com/ytdl-org/youtube-dl/issues/30568 .

2.15.11. See https://phihag.de/keys/A4826A18.asc .

2.15.3.3 Binary signing key. PHILIPP HAGEMEISTER (DB4B 54CB A482 6A18)

A binary signing key used used by Philipp Hagemeister to sign releases sometime before 2021.

```
pub   rsa4096/0xDB4B54CBA4826A18 2013-01-11 [SC] [expires: 2033-01-06]
      Key fingerprint = 7D33 D762 FD6C 3513 0481  347F DB4B 54CB A482 6A18
uid                   [ unknown] Philipp Hagemeister <9482dbb6>
uid                   [ unknown] Philipp Hagemeister <3ec335f6>
uid                   [ unknown] Philipp Hagemeister <7787a9ff>
sub   rsa4096/0x862A257D825E38B8 2013-01-11 [E] [expires: 2033-01-06]
      Key fingerprint = 61F8 AC9E 8A81 6A5F 9BD8  B922 862A 257D 825E 38B8
```

2.15.3.4 Binary signing key. FILIPPO VALSORDA (EBF0 1804 BCF0 5F6B)

A binary signing key used by Filippo Valsorda to sign releases sometime before 2021.

```
pub   rsa4096/0xEBF01804BCF05F6B 2012-08-30 [SCEA]
      Key fingerprint = 428D F5D6 3EF0 7494 BB45  5AC0 EBF0 1804 BCF0 5F6B
uid                   [ unknown] Filippo Valsorda <7970bea1>
uid                   [ unknown] Filippo Valsorda <651b1dcc>
uid                   [ unknown] Filippo Valsorda <1b03dbe9>
```

Appendix A
How to use GNUPG

Last updated 2022-05-18 by STEVEN BALTAKATEI SANDOVAL.

This appendix describes in more detail how to use GNUPG. Examples assume use of GNUPG version *v2.2.12*. Definitions of terms relevant to GNUPG are provided in A.1. Useful commands are provided in A.2.

Remark A.0.1. Example code is sometimes given in the form of a BASH *script*. Such scripts usually have a first line like `#!/usr/bin/env bash` that tell your interpreter to execute the lines that follow as BASH commands. This is useful from a typography standpoint because often the length of GNUPG commands can exceed the recommended character limit for human readability.[A.0.1] This document will attempt to limit line widths in code examples to approximately 80 characters.

A.1 Terms and Definitions

authenticate.

1. (verb) An operation performed by Alice to authenticate herself to a server (e.g. to open a command line interface on a remote server via `ssh`)[A.1.1].
2. (noun) A *capability flag* on a *primary key* or *subkey* indicating that the key may be used to perform the *authenticate* operation.

capability. (adjective) See *flag*.

capability flag. (noun) See *flag*.

certify.

1. (verb) An operation performed by Alice's *primary key* (a type of *public key*) on a *User ID* of Bob's primary key to indicate that Alice believes Bob's primary key is actually his.
2. (noun) A *capability flag* on a *primary key* or *subkey* indicating that the key may be used to perform the *certify* operation.

decrypt. (verb) The action a *primary key* or *subkey* with an *encrypt* flag performs on an encrypted file.

encrypt.

1. (verb) An operation performed by Alice on a file against a *primary key* or *subkey* owned by Bob in order to send Bob the file without anyone else being able to read the file plaintext.
2. (noun) A *capability flag* on a *primary key* or *subkey* indicating that the key may be used to perform the *encrypt* operation which decrypts[A.1.2] the file.

encrypted. (adjective) A state of a file or data in which the contents are unreadable except by someone possessing a *private key* or *symmetric key*. Such keys can *decrypt* the contents to reveal the original plaintext.

A.0.1. EMACS, for example, defaults to wrapping columns of text to 70. See `https://emacs.stackexchange.com/questions/36118/` .

A.1.1. See `https://mlohr.com/gpg-agent-for-ssh-authentication-update/` . Accessed 2022-05-17.

A.1.2. See *decrypt*.

fingerprint. (noun) A number, usually expressed in *hexadecimal*, that uniquely identifies a *public key*. In GNUPG, this usually refers to the *full fingerprint* which takes the form of a 40-character string (ignoring spaces) that may be displayed using `gpg --fingerprint key-id` (e.g. `3457 A265 922A 1F38 39DB  0264 A0A2 95AB DC34 69C9`). The string is derived from a cryptographic digest of the *public key*.[A.1.3] The *long ID* is a substring of the *full fingerprint*.

flag. (noun) (a.k.a. "*capability flag*") A small digital marker on a *public key* indicating to OPENPGP software how the key should be used. Possible capability flags (and their abbreviations) include: *certify* (`C`), *sign* (`S`), *encrypt* (`E`), and *authenticate* (`A`).

Flags are set during key creation (e.g. via `gpg --expert --full-gen-key`) and may be modified later (e.g. via `gpg --edit-key key-id`). A key generated using completely default settings with GNUPG *v2.2.12* will consist of a *primary key* with *sign* (`S`) and *certify* (`C`) capability flags and a single *subkey* with an *encrypt* (`E`) capability flag.

full fingerprint. (noun) In GNUPG, a 40-character string encoding a 160-bit number derived from a cryptographic digest of a *public key*. See *fingerprint*.

hexadecimal.

1. (countable) A number expressed in base 16 notation (e.g. "`7155`" is "`0x1BF3`" in hexadecimal).
2. (adjective) A property of a number that is expressed in hexadecimal.

interactive. (adjective) A property of a method that provides a program input which requires the full attention of a user. Contrast with *non-interactive*. For example, when GNUPG prompts the user to enter a passphrase and a user types the passphrase using their keyboard.

key. (noun) An abstract object that can be used to digitally sign or decrypt data. Categories of keys used in GNUPG include *public key*, *private key*, and *symmetric key*. A public key may be marked to be used for use as a *primary key* or a *subkey*. Depending upon context, *key* may consist of multiple such objects (e.g. a *primary key* and associated *subkey*(s) are modified by `gpg --edit-key`).

keybox. (noun) A file format used by GNUPG for storing public keys.[A.1.4] See *keyring*.

key-id. (noun) A string of characters that identifies a *public key* such as a *long ID* or *fingerprint*.

keypair. (noun) A *public key* and its associated *private key*.

keyring. (noun) A set of keys that can be modified using `gpg --edit-key`. Since GNUPG *v2.1*, public keys are stored by default in a *keybox* file at `.gnupg/pubring.kbx`.[A.1.5]

keystore. (noun) (alt. "key store") A term not used by GNUPG but which may be refer to a collection of *key*s that GNUPG would call a *keyring*.

long ID. (noun) A 16-digit *hexadecimal* number used to identify a *public key*, e.g. `A0A2 95AB DC34 69C9`. Its hexadecimal nature may be emphasized by prepending the string with the "`0x`" prefix and omitting spaces, e.g. `0xA0A295ABDC3469C9`.[A.1.6] GNUPG is not particular about whether letters in the *long ID* are upper or lowercase, so `0xa0a295abdc3469c9` is also acceptable. Compare with *short ID*.

non-interactive. (adjective) A property of a method that provides a program input which does not require the full attention of a user. Contrast with *interactive*. For example, when a script uses the `--batch --yes --passphrase string` options in order to automatically provide a passphrase `string` to gpg unattended, the script may be described as *non-interactive*.

primary key. (noun) In GNUPG, a *public key* or *keypair* that is generally used to *certify* one's own *subkey* or another person's *primary key*. The *key* may be marked by some combination of *certify*, *sign*, *encrypt*, or *authenticate* capability flags. A *primary key* has a uniquely identifying *fingerprint*. See *long ID*.

A.1.3. See https://blog.djoproject.net/2020/05/03/main-differences-between-a-gnupg-fingerprint-a-ssh-fingerprint-and-a-keygrip/ .

A.1.4. See https://www.gnupg.org/documentation/manuals/gnupg/kbxutil.html .

A.1.5. See https://gnupg.org/faq/whats-new-in-2.1.html .

A.1.6. See https://stackoverflow.com/questions/2670639/why-are-hexadecimal-numbers-prefixed-with-0x .

primary UID. (noun) The main *UID* (*User ID*) to be used when multiple *UID*s are present. May be set for key A0A2 95AB DC34 69C9 with:

```
gpg --quick-set-primary-uid 0xa0a295abdc3469c9 primary-user-id
```

primary user ID. (noun) See *primary UID*.

plaintext. (noun) The state of data or a file before it is encrypted against a *public key* or a *symmetric key*.

private key. (noun) A *key* that is used with another user's *public key* to encrypt data or sign data. Together with a *public key*, forms a *keypair*.

public key. (noun) A *key* that is used with another user's *private key* to decrypt data they encrypted or verify data they signed. Together with a *private key*, forms a *keypair*. May be uniquely identified with a *fingerprint*.

script. (noun) An executable file that may run programs such as gpg. Often used for the purpose of automating complex tasks such as encrypting many files at once.

sign. (verb) A *private key* operation that produces a *signature*.

signature. (noun) A unique cryptographic proof generated by with a *private key* against some data (e.g. a file) that indicates the owner of the private key possessed a copy of the data. Often used to indicate validity of important documents or executables (e.g. verifying an `.iso` file used to install an operating system was not corrupted or compromised by an attacker). Also known as a "digital signature".

In GNUPG, a signature of `file.txt` may be created and stored in `file.txt.gpg` with:

```
$ gpg --detach-sign --output file.txt.gpg -- file.txt
```

subkey. (noun) In GNUPG, a *public key* or *keypair* that is generally used to perform operations in the place of a *primary key* in situations where the risk of leaking a *primary key*'s *private key* is unacceptable (e.g. on a computer that could be physically stolen or remotely hacked). A *subkey* can be made mostly functionally equivalent to a *primary key* except for hidden software indicators identifying it as a *subkey*. It is discouraged to assign *certify* capability to a *subkey*.[A.1.7]

symmetric key. (noun) A *key* that is used to both encrypt and decrypt. In GNUPG, a symmetric key consists of a passphrase that may be provided interactively or non-interactively. For example, `file.txt` may be encrypted using the symmetric key "1234" using:

```
gpg --symmetric --output file.txt.gpg -- file.txt
```

short ID. (noun) An 8-digit *hexadecimal* number similar to a *long ID*. Use of *short ID* is not recommended because, as of 2021, generating multiple public keys with matching *short ID*s requires a negligible amount of computing power.[A.1.8] Compare with *long ID*.

UID. (noun) Abbreviation of *User ID*.

User ID. (noun) (a.k.a. "*UID*"). Identification data for a *primary key*. Usually consists of a name and email address. "*User ID*" stands for "User Identification" and may be shortened to "*UID*". When two people sign eachother's PGP keys, what is meant is that each uses a *certify*-capable *key* (usually their *primary key*) to sign one or more of eachother's *User ID*s . Unusually, a *User ID* may consist of a JPEG image file.

GNUPG may identify a *User ID* through partial or exact string matches.[A.1.9]

See *UID*. See also *primary UID*.

--. (command line syntax) Indicates the end of options passed to a command and the starting position of positional arguments. Also known as a "double dash". Used by many other command line tools such as `bash` or `grep`. May be useful to more clearly indicate which arguments are NOT associated with an option flag. For example, the following commands are functionally equivalent but the latter more clearly indicates that `file.txt` is the only non-option argument which gpg interprets as the input:

```
gpg --encrypt --output file.txt.gpg file.txt
```

A.1.7. See https://web.archive.org/web/20220517224040/https://lists.gnupg.org/pipermail/gnupg-users/2017-August/058904.html .

A.1.8. See https://security.stackexchange.com/questions/84280/ .

A.1.9. See https://www.gnupg.org/documentation/manuals/gnupg/Specify-a-User-ID.html .

`gpg --encrypt --output file.txt.gpg -- file.txt`

Below is a `grep` example in which a naïve search of `file.txt` for the string `-violet-` fails without the double dash:

```
$ echo "asdf-violet-asdf" > file.txt
$ grep --only-matching "-violet-" file.txt
$ grep --only-matching -- "-violet-" file.txt
-violet-
```

A.2 Useful Commands

A.2.1 Obtaining keys

A.2.1.1 Import a public key

The `$ gpg --import -- key.asc` command may be used to import a file named "`key.asc`". If the `$ gpg --import` command by itself is run and a clipboard program is available (e.g. copy/paste), then pasting the text of a public key into the shell followed by pressing `ctrl-d` (i.e. providing an "end of transmission" character[A.2.1]) will tell gpg to process the pasted text.

A.2.1.2 Download from a keyserver

The `$ gpg --receive-keys` command can be used as shown in the example below to download a public key (e.g. 4246 8F40 09EA 8AC3) from a keyserver (e.g. keyserver.ubuntu.com).

```
$ gpg --receive-keys --keyserver keyserver.ubuntu.com -- 42468f4009ea8ac3
gpg: key 0x42468F4009EA8AC3: public key "Debian Testing CDs Automatic... <047be9f4>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

As of 2022-01-14, few keyservers provide full public keys due to an unsolved certificate spam problem.[A.2.2]

- keyserver.ubuntu.com - Provides full keys.
- keyring.debian.org - Provides full keys of DEBIAN developer and maintainers.
- keys.openpgp.org - Provides keys without user IDs unless key owner authenticates themselves via the user ID email address.

A.2.2 Analyzing keys

A.2.2.1 View public key fingerprint

- Show *fingerprint*s of the *primary key* and *subkey*s . The example below shows the primary fingerprint in red, the *long ID* colored in brown, *User ID*s in blue[A.2.3], and fingerprints of subkeys dark green.

```
$ gpg --fingerprint -- 0xa0a295abdc3469c9
pub   rsa4096/0xA0A295ABDC3469C9 2017-10-11 [C] [expires: 2022-07-08]
      Key fingerprint = 3457 A265 922A 1F38 39DB  0264 A0A2 95AB DC34 69C9
uid                   [ultimate] Steven Sandoval <baltakatei@gmail.com>
uid                   [ultimate] Steven Sandoval <baltakatei@alumni.stanford.edu>
sub   rsa4096/0x6DD7D496916A1253 2018-05-16 [E] [expires: 2022-07-07]
      Key fingerprint = 5E55 5FC6 1C85 871E 813B  5BCF 6DD7 D496 916A 1253
sub   rsa4096/0x57DA57D9517E6F86 2018-05-16 [S] [expires: 2022-07-07]
      Key fingerprint = 38F9 6437 C83A C88E 28B7  A952 57DA 57D9 517E 6F86
sub   rsa4096/0x5F9D26B9A598A2D3 2018-05-16 [A] [expires: 2022-07-07]
      Key fingerprint = EDCA 7EE7 D09E 7F2E 1DF6  A229 5F9D 26B9 A598 A2D3
```

A.2.1. See https://unix.stackexchange.com/a/110248 .

A.2.2. Hansen, Robert J.. "SKS Keyserver Network Under Attack". 2019-06-29. https://gist.github.com/rjhansen/67ab921ffb4084c865b3618d6955275f .

A.2.3. In this publication, I may obfuscate email addresses via a `b2sum -l32` hash if I cannot confirm the key owner is okay with their email address being posted so plainly.

A.2.3 Sending keys

A.2.3.1 Export public key

- Export *public key* according to last 16 characters of public key *fingerprint* (i.e. "*long ID*", e.g. A0A2 95AB DC34 69C9).

```
$ gpg --export --output /tmp/key -- 0xa0a295abdc3469c9.
```

- Export the smallest key possible. Useful to strip key of signatures except for self-signatures. This creates an ASCII-armored[A.2.4] text file named `pubkey.asc` in the `/tmp` directory.

```
#!/usr/bin/env bash
gpg --export --export-options export-minimal \
    --armor \
    --output /tmp/pubkey.asc \
    -- \
    0xa0a295abdc3469c9
```

A.2.3.2 Upload public key

- Send *public key* to a keyserver using gpg and a *long ID*.

```
$ gpg --send-keys --keyserver keyserver.ubuntu.com -- 0xa0a295abdc3469c9
```

  - Note: Some keyservers such as `keys.openpgp.org` achieve reliability by requiring uploaders to verify their identity via an email exchange through a *User ID* email address. If no verification is performed for a given *User ID*, uploaded keys are shared without that *User ID*.[A.2.5]

- Send *public key* to the `https://keys.openpgp.org` keyserver using a web browser.
  - Export a public key file (see A.2.3.1).
  - Go to `https://keys.openpgp.org/upload` and submit the public key file.
  - Go to `https://keys.openpgp.org/manage` and submit the email address of the public key's main *User ID*.
  - Verify the *User ID* by following the emailed instructions sent by `https://keys.openpgp.org` .

A.2.4 Creating keys

A.2.4.1 Using default settings

Running `$ gpg --gen-key` will guide the user to creating a key with default settings.

A.2.4.2 With subkeys

The `$ gpg --expert --full-gen-key` command in combination with some modifications to the configuration file `~/.gnupg/gpg.conf` may be used to create an OpenPGP key with subkeys. Subkeys are useful since their private components can be loaded onto a smartcard while keeping the primary key offline, available to create new subkeys. This may be desireable if a primary key is intended to be used over a long time period and the risk of losing an online defaultly configured key is unacceptable. Please see the article by THIERRY THURON titled "OpenPGP - The Almost Perfect Key Pair" for a useful procedure.[A.2.6]

A.2.4. See `https://crypto.stackexchange.com/questions/91984/why-use-ascii-armor-for-file-encryption` .

A.2.5. See `https://keys.openpgp.org/about` .

A.2.6. Thuron, Thierry. "OpenPGP - The Almost Perfect Key Pair". 2017-10-13. Eleven Labs Blog. `https://blog.eleven-labs.com/en/openpgp-almost-perfect-key-pair-part-1/` .

Bibliography

[1] Jim Dwyer. Decentralizing the Internet So Big Brother Can't Find You. *New York Times*, February 2018.
[2] Steve Lohr. Microsoft Buys GitHub for $7.5 Billion, Moving to Grow in Coding's New Era. *New York Times*, June 2018.

Index