Notable Public Keys

STEVEN BALTAKATEI SANDOVAL

Email: baltakatei@gmail.com

Web: https://reboil.com

Front Matter

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

https://reboil.com/gitweb/BK-2021-09.git

This book was typeset using **TEXMACS** version `2.1.1`.
Fonts used include **Linux Libertine**.
This book was rendered on `2022-03-06T12:10:32+0000`.

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

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 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.

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 were revoked by other BITCOIN CORE developers after Andresen published a blog post claiming Craig Wright was Satoshi Nakamoto.[2.1.8]

2.1.3 Public Key Details

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

This key[2.1.9], owned by WLADIMIR J. VAN DER LAAN, has been used to sign BITCOIN CORE releases since version `0.11.0`.

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

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

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

WLADIMIR VAN DER LAAN used his personal key[2.1.10] to sign BITCOIN versions v0.9.3–v0.10.2.

```
pub   rsa2048/0x74810B012346C9A6 2011-08-24 [SC] [expires: 2022-02-10]
      Key fingerprint = 71A3 B167 3540 5025 D447  E8F2 7481 0B01 2346 C9A6
uid                   [ unknown] Wladimir J. van der Laan <laanwj@visucore.com>
uid                   [ unknown] Wladimir J. van der Laan <laanwj@gmail.com>
uid                   [ unknown] Wladimir J. van der Laan <laanwj@protonmail.com>
sub   rsa2048/0x69B4C4CDC628F8F9 2017-05-17 [A] [expires: 2022-02-10]
sub   rsa2048/0xF69705ED890DE427 2011-08-24 [E]
sub   rsa2048/0x1E4AED62986CD25D 2017-05-17 [S] [expires: 2022-02-10]
```

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

Gavin Andresen used this dedicated code-signing key[2.1.11] 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/ .

```
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) <gavinandresen@gmail.com>
sub   rsa4096/0x1B7BFB457BF6E212 2013-11-01 [S]
sub   rsa4096/0x36E924A98E30B3ED 2011-12-15 [E]
```

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

The dsa1024 algorithm this key[2.1.12] uses is considered weak by the the NIST standard SP800-57 Part 1 Revision 5: *Recommendation for Key management*.[2.1.13] 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[2.1.14]. 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 <satoshin@gmx.com>
sig 3        0x18C09E865EC948A1 2008-10-30  Satoshi Nakamoto <satoshin@gmx.com>
sig          0x74810B012346C9A6 2013-05-10  Wladimir J. van der Laan <laanwj@visucore.com>
sig 1        0x7FAB114267E4FA04 2013-10-12  Peter Todd <pete@petertodd.org>
rev          0x74810B012346C9A6 2016-05-02  Wladimir J. van der Laan <laanwj@visucore.com>
      reason for revocation: User ID is no longer valid
sub   elg2048/0xCF1857E6D6AAA69F 2008-10-30 [E]
sig          0x18C09E865EC948A1 2008-10-30  Satoshi Nakamoto <satoshin@gmx.com>
```

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

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

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

2.1.13. 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.

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

2.2 CRYPTOMATOR

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 (0x69CEFAD519598989).

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] [expires: 2021-12-31]
      Key fingerprint = 5054 3A3D A4B1 DB81 DA3E  79CB 509C 9D63 34C8 0F11
uid                   [ unknown] Cryptobot (Release Manager) <releases@cryptomator.org>
```

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).

```
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 <releases@cryptomator.org>
```

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 <releases@cryptomator.org>
sig 3        0x615D449FE6E6A235 2020-08-18  Cryptobot <releases@cryptomator.org>
sig          0x667B866EA8240A09 2020-08-18  [User ID not found]
sig          0x509C9D6334C80F11 2020-09-02  Cryptobot (Release Manager) <releases@cryptomator.org>
```

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

2.3 DEBIAN

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, 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 <steve@einval.com>
uid                   [ revoked] Steve McIntyre <stevem@chiark.greenend.org.uk>
uid                   [ revoked] Steve McIntyre <93sam@debian.org>
uid                   [ revoked] Debian CD signing key <debian-cd@lists.debian.org>
```

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) <manty@debian.org>
uid                   [ unknown] Santiago Garcia Mantinan (manty) <sgm@manty.net>
uid                   [ unknown] Santiago Garcia Mantinan (manty) <manty@gpul.org>
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 <mail@daniel-baumann.ch>
```

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 <daniel@127011.net>
uid                   [ unknown] Daniel Baumann <daniel@undebian.org>
uid                   [ unknown] Daniel Baumann <daniel@debian-unofficial.org>
uid                   [ unknown] Daniel Baumann <daniel@unable-to-package.org>
uid                   [ unknown] Daniel Baumann <daniel.baumann@panthera-systems.net>
uid                   [ unknown] Daniel Baumann <daniel@free-law.ch>
uid                   [ unknown] Daniel Baumann <mail@daniel-baumann.ch>
uid                   [ unknown] Daniel Baumann <daniel@debian.org>
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 <debian-cd@lists.debian.org>
```

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 <debian-cd@lists.debian.org>
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 <debian-live@lists.debian.org>
```

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) <debian-live@lists.debian.org>
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 <debian-live@lists.debian.org>
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 <debian-cd@lists.debian.org>
sub   rsa4096/0x0C5470136BD05CFB 2014-04-15 [E]
      Key fingerprint = AEE9 9CA7 0C3E C4B3 1C75  2843 0C54 7013 6BD0 5CFB
```

2.4 GitHub

2.4.1 Background

GitHub[2.4.1] is a commercial Git repository hosting service company founded in 2008. It was purchased by Microsoft in 2016.[1]

2.4.2 History

2008. GitHub founded in San Francisco.[1]

2008-03-10. GitHub parent company Logical Awesome, LLC registered in San Francisco by Chris Wanstrath.[2.4.2]

2008-05-14. First snapshot of the `https://github.com` website on the Internet Archive.[2.4.3]

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.[2.4.4] Also the date of a post by GitHub user `jonathancross`[2.4.5] observing that the `4AEE 18F8 3AFD EB23` key appears to be a new feature[2.4.6]:

> 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.[2.4.7]

2021-05-25. Public key `4AEE 18F8 3AFD EB23` fingerprint explicitly published at GitHub documentation website.[2.4.8]

2.4.3 Public Key Details

2.4.3.1 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[2.4.9] 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) <noreply@github.com>
```

This key is available for download at GitHub's documentation website at `https://github.com/web-flow.gpg` .[2.4.10] This particular link as well as the full key fingerprint was added to the GitHub documentation repository in a commit dated 2021-05-25[2.4.11].

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

2.4.2. 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` .

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

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

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

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

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

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

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

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

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

2.5 RASPIBLITZ

2.5.1 Background

RASPIBLITZ[2.5.1] 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.[2.5.2] As of 2021-07-18, the principal maintainer appears to be CHRISTIAN "ROOTZOL" ROTZOLL[2.5.3].

2.5.2 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.[2.5.4] 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.[2.5.5]

2021-02-07. Andreas Antonopoulos posted a YouTube video identifying RASPIBLITZ as a popular Bitcoin full node software package.[2.5.6]

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

2.5.3 Public Key Details

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

ROOTZOL's PGP key[2.5.7] may be downloaded from their Keybase page.[2.5.8]. 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 <christian@rotzoll.de>
sub   rsa4096/0xAA9DD1B5CC5647DA 2019-09-03 [S] [expires: 2021-10-21]
sub   rsa4096/0xD40D94E6C7C9B4D9 2019-09-03 [E] [expires: 2021-10-21]
sub   rsa4096/0x1C29DC2F8D764F9A 2019-09-03 [A] [expires: 2021-10-21]
```

2.5.1. Main website: `https://raspiblitz.org/` .

2.5.2. See `https://github.com/rootzoll/raspiblitz` .

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

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

2.5.5. See `https://web.archive.org/web/20201031223643/https://raspiblitz.org/` .

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

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

2.5.8. See `https://keybase.io/rootzoll/pgp_keys.asc` .

2.6 SATOSHI LABS

2.6.1 Background

SATOSHI LABS[2.6.1] is a company that produces cryptocurrency hardware wallets called TREZOR[2.6.2]. 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.[2.6.3] It is based in Prague, Czech Republic.

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

2.6.2 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.[2.6.4]

2017-01-11. mytrezor.com , buytrezor.com , and other domains migrated to https://trezor.io .[2.6.5]

2017-01-28. The first snapshot of https://trezor.io appears on the INTERNET ARCHIVE.[2.6.6]

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[2.6.7] in order to replace an older web wallet implementation.[2.6.8]

2.6.3 Public Key Details

2.6.3.1 PAVOL RUSNÁK (91F3 B339 B9A0 2A3D)

A key[2.6.9] used by a developer named PAVOL "STICK" RUSNÁK.[2.6.10] This key has been used to sign TREZOR software in the past[2.6.11] such as TREZOR BRIDGE[2.6.12] and other various GITHUB commits.

```
pub   rsa4096/0x91F3B339B9A02A3D 2012-03-07 [SC] [expires: 2022-01-16]
      Key fingerprint = 86E6 792F C27B FD47 8860  C110 91F3 B339 B9A0 2A3D
uid                   [ unknown] Pavol Rusnák <pavol@rusnak.io>
uid                   [ unknown] Pavol Rusnák <stick@gk2.sk>
uid                   [ unknown] Pavol Rusnák <prusnak@opensuse.org>
uid                   [ unknown] Pavol Rusnák <stick@satoshilabs.com>
uid                   [ unknown] [jpeg image of size 2449]
sub   rsa4096/0x22AF226D38DC1F4D 2012-03-07 [E] [expires: 2023-01-08]
      Key fingerprint = E177 6F65 0601 E596 9E7F  9E25 22AF 226D 38DC 1F4D
```

2.6.3.2 2020 Signing Key (26A3 A566 62F0 E7E2)

A key[2.6.13] used to sign the software required by a PC to communicate with the TREZOR product line. Expired as of 2021-01-01.

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

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

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

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

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

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

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

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

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

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

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

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

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

```
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
```

2.6.3.3 2021 Signing Key (E21B 6950 A2EC B65C)

A key[2.6.14] 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
```

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

2.7 TOR BROWSER

2.7.1 Background

TOR BROWSER[2.7.1] is a browser software package that permits visiting websites with anonymity effected by onion routing. Although various[2.7.2] 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.

2.7.2 History

2008-01-30. STEVEN J. MURDOCH announces development of TOR BROWSER.[2.7.3]

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.[2.7.4] Other high-profile PGP keys were also affected at this time.[2.7.5]

2.7.3 Public Key Details

2.7.3.1 Release Signing Key (4E2C 6E87 9329 8290)

Public key used for signing TOR BROWSER releases since at least 2015-03-15[2.7.6] until 2022-03-06[2.7.7].

```
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) <torbrowser@torproject.org>
sub   rsa4096/0xE53D989A9E2D47BF 2021-09-17 [S] [expires: 2023-09-17]
      Key fingerprint = 6131 88FC 5BE2 176E 3ED5  4901 E53D 989A 9E2D 47BF
```

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

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

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

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

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

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

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

2.8 YOUTUBE-DL

2.8.1 Background

YOUTUBE-DL[2.8.1] is a PYTHON2-based[2.8.2] 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.[2.8.3]

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

Since 2021-12-25, the core developer is REMITA AMINE[2.8.4] (?).

2.8.2 History

2008-07-21. First commit in the main project GIT repository published by RICARDO GARCIA.[2.8.5]

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[2.8.6] issued by the RIAA.[2.8.7]

2020-11-16. GITHUB page for YOUTUBE-DL reinstated.[2.8.8]

2021-12-25. The only active developer is REMITA AMINE (?).[2.8.9]

2022-01-29. The project announced[2.8.10] 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.

2.8.3 Public Key Details

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

The binary signing key used to sign releases as of 2021.

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

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

A binary signing key used by Philipp Hagemeister to sign releases sometime before 2021.[2.8.11]

```
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 <ubuntu@phihag.de>
uid                   [ expired] Philipp Hagemeister <phihag@phihag.de>
```

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

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

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

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

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

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

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

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

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

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

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

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

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

```
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 <phihag@phihag.de>
uid                   [ unknown] Philipp Hagemeister <philipp.hagemeister@uni-duesseldorf.de>
uid                   [ unknown] Philipp Hagemeister <hagemeister@cs.uni-duesseldorf.de>
sub   rsa4096/0x862A257D825E38B8 2013-01-11 [E] [expires: 2033-01-06]
      Key fingerprint = 61F8 AC9E 8A81 6A5F 9BD8  B922 862A 257D 825E 38B8
```

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

A binary signing key used by ⟨index|Valsorda, Filippo⟩ 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 <fv@filippo.io>
uid                   [ unknown] Filippo Valsorda <filippo.valsorda@gmail.com>
uid                   [ unknown] Filippo Valsorda <filosottile.wiki@gmail.com>
```

Appendix A
How Public Key Cryptography Works

This appendix describes in more detail how public key cryptography works.

Appendix B
How to use GNUPG

This appendix describes in more detail how to use GNUPG. Examples assume use of GNUPG version *2.2.12*.

B.1 Definitions

Long ID. 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`.[B.1.1] GNUPG is not particular about whether letters in the Long ID are upper or lowercase, so `0xa0a295abdc3469c9` is also acceptable.

Short ID. An 8-digit hexadecimal number similar to a Long ID. Use of Short IDs is not recommended because, as of 2021, generating multiple public keys with matching Short IDs requires a negligible amount of computing power.[B.1.2]

Remark B.1.1. 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.[B.1.3] This document will attempt to limit line widths in code examples to approximately 80 characters.

B.2 Useful Commands

B.2.1 Obtaining keys

B.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[B.2.1]) will tell `gpg` to process the pasted text.

B.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 ... <debian-cd@lists.debian.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

B.1.1. See https://stackoverflow.com/questions/2670639/why-are-hexadecimal-numbers-prefixed-with-0x .

B.1.2. See https://security.stackexchange.com/questions/84280/ .

B.1.3. EMACS, for example, defaults to wrapping columns of text to 70. See https://emacs.stackexchange.com/questions/36118/ .

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

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

B.2.2 Analyzing keys

B.2.2.1 View public key fingerprint

- Show fingerprints of the primary key and subkeys . The example below shows the primary fingerprint in red, the Long ID colored in brown, user IDs in blue, 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
```

B.2.3 Sending keys

B.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[B.2.3] 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
```

B.2.4 Creating keys

B.2.4.1 Using default settings

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

B.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.[B.2.4]

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

B.2.3. See https://crypto.stackexchange.com/questions/91984/why-use-ascii-armor-for-file-encryption .

B.2.4. 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] Steve Lohr . Microsoft Buys GitHub for $7.5 Billion, Moving to Grow in Coding's New Era. *New York Times*, 2018.

Index