

OpenPGP Email Encryption

Workshop



by Alex Chesnakov

About me

- Alex Chesnakov 
- Senior Engineer, Security Manager @ Health Gorilla
- Security
- Linux, AWS, CTFs
- You may see me sailing or flying kites at Shoreline Lake

All opinions are my own and not those of my employer.

Goals

- **Introduction to OpenPGP encryption**
- **Create OpenPGP keys**
- **Setup OpenPGP encryption in Thunderbird email client**
- **Exchange encrypted email messages with 2-3 recipients**
- **Encrypt, Decrypt and Sign text / files**

Encryption

Encryption Algorithms:

- Symmetric (AES256, encrypted zip file)
- Asymmetric (OpenPGP)

Usage:

- In transit (<https://>, WIFI encryption)
- At Rest (Mac and Windows disk encryption)
- End-to-End Encryption (OpenPGP)

Why

- [**Before 1996 - Illegal to export**](#)
 - [**Encryption ban proposal in the UK**](#)
 - [**Encryption ban proposal in EU**](#)
 - [**Encryption ban proposal in the US**](#)
-
- I want to encrypt a few files
 - No encryption necessary, I just want to be sure email indeed comes from the person
 - I found a 0-day but company insists I use their OpenPGP to encrypt email before sending
 - I downloaded .iso or program, they suggest to check signature, but I don't know how



Terminology

- PGP - Pretty Good Privacy Software. Proprietary, PGP Corp -> Symantec -> Broadcom
- GPG - GNU Privacy Guard Software. Free and open-source
- OpenPGP – Standard. PGP and GPG are OpenPGP-compliant
- Plaintext - “*Secret Message*” or binary file, like photo
- Ciphertext - “*Frperg Zrffntr*” encrypted text
- Cipher - algorithm for performing encryption/decryption. ROT13: A-N, B-O, C-P
- Hash - function map data of arbitrary size to fixed-size value output
 - Hash always has the same length for same hash algorithm (MD5, SHA256)
 - Change one bit in input, 50% different bits in hash
 - The same input always gives the same hash
 - One way function - cannot recover plaintext from a hash

Goals of PGP Cryptography

- **Confidentiality** - message contents remain private
- **Integrity** - refers to keeping a message unchanged
- **Nonrepudiation** - person cannot deny signing a particular message
- **Authenticity** - All these effects occurring simultaneously.

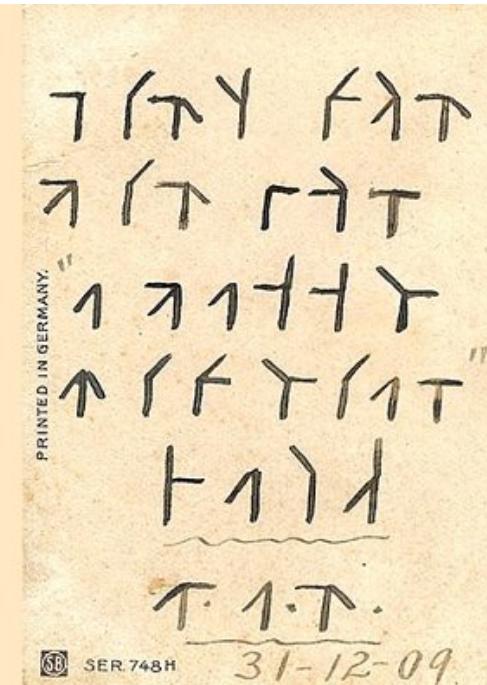
Main Usage Examples

Goal	How
I want anyone who reads this message to be sure I sent it and it was not changed (Nonrepudiation, Integrity)	Digitally sign the message with your private key
I want to verify the sender's identity	Verify signature with the sender's public key.
I want to send a message that only my intended recipient can read (Confidentiality)	Encrypt the message with the recipient's public key
I want to decrypt a message that I received	Decrypt the message with your private key
I want my message to be readable only by recipient, and I want recipient to be sure message came from me (Authenticity)	Encrypt the message with the recipient's public key and digitally sign the message with your private key
I want to decrypt and verify a message that includes a digital signature	Decrypt the message with your private key and verify the signature with the sender's public key

Confidentiality

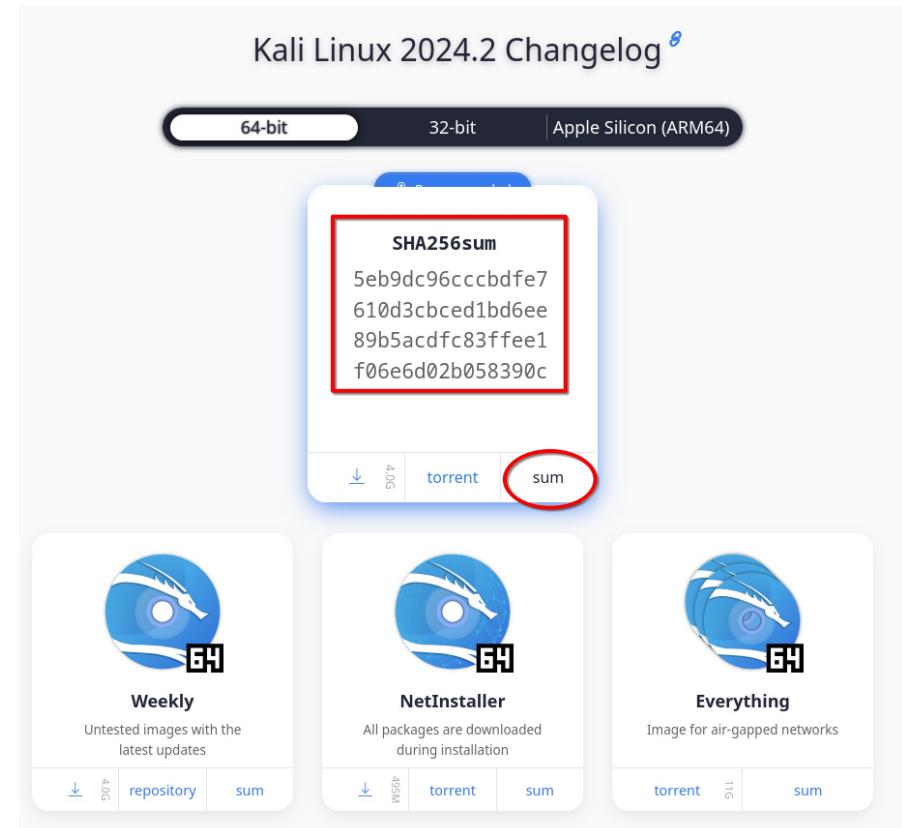
Confidentiality means that the message contents remain private. The *plaintext* cannot be viewed by anyone who doesn't have the necessary keys.

In many cases, you cannot prevent someone from viewing the *ciphertext*, because every message over the Internet can be viewed by a large number of people, just as postcards can be viewed by a postman or a neighbor.



Integrity

Integrity refers to keeping a message unchanged. By using OpenPGP, you can confirm that a message has not been tampered with during transmission.

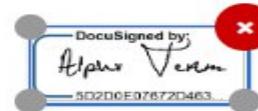


Nonrepudiation

Nonrepudiation means that a person cannot deny signing a particular message, which is especially important in the context of email.

Example of nonrepudiation:

- Software vendors **Sign** program, drivers, .iso files
- DocuSign



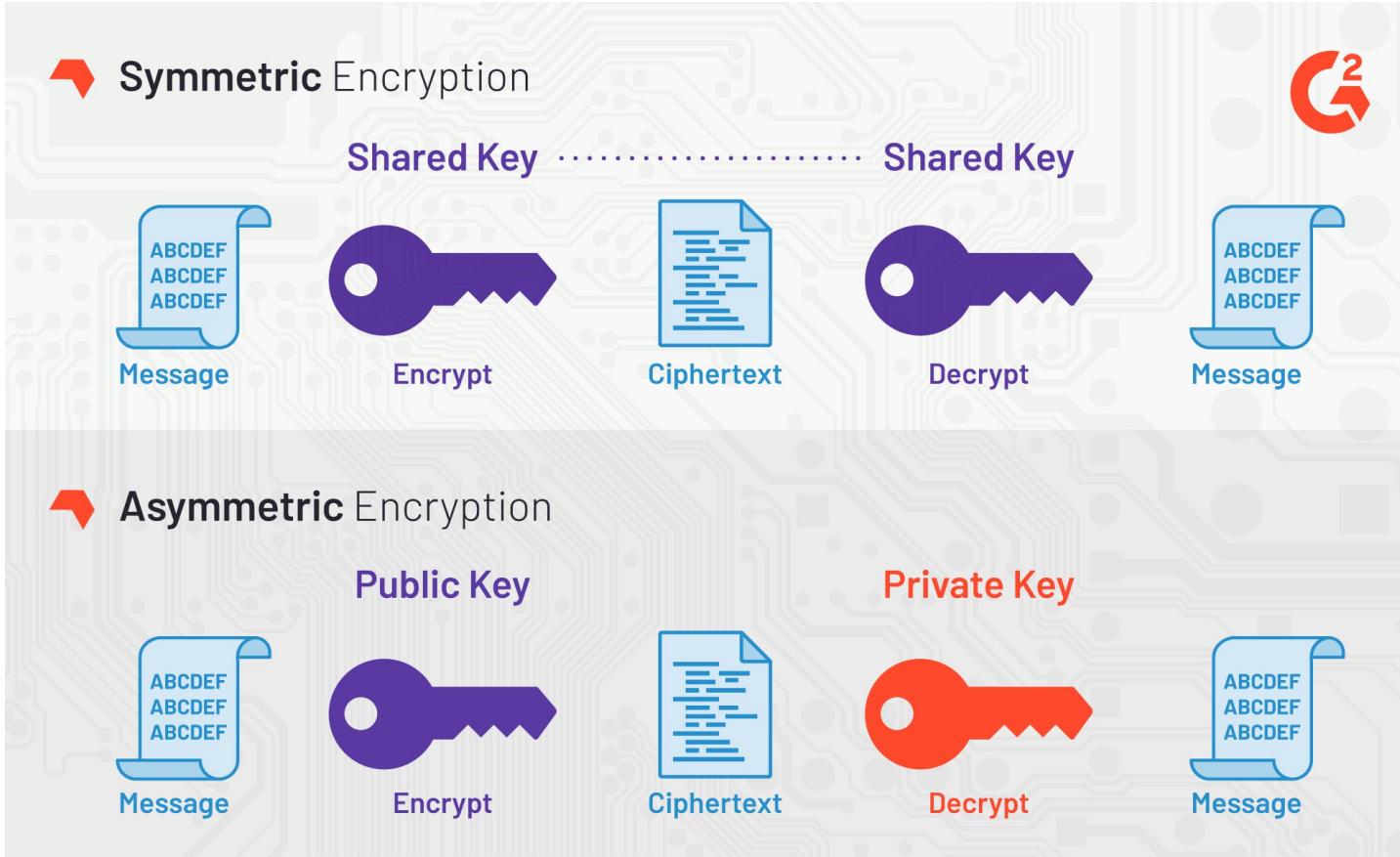
DocuSign®

Authenticity

When you receive an email that has been **encrypted** and **signed** with OpenPGP, you know that the contents of the message have been **concealed** from any eavesdroppers. You know that the content of the message has **not** been **changed**. You also know that the message **comes** from the **person** who has the right to send such a message in the sender's name

The screenshot shows a message security interface for OpenPGP. At the top right, there is a header with the text "OpenPGP" followed by icons for a lock and a gear. Below this, the main title is "Message Security - OpenPGP". There are two main sections: "Good Digital Signature" and "Message Is Encrypted".
Good Digital Signature: This section indicates that the message includes a valid digital signature from a verified key. It shows the signer's key ID: "Signer key ID: 0x9FF2194CC09A61E8" and a "View signer key" button.
Message Is Encrypted: This section indicates that the message was encrypted before it was sent to the recipient. It ensures that the message can only be read by the intended recipients. It shows the decryption key ID: "Your decryption key ID: 0x1FA31C2070C6150F (Sub key ID: 0x1492A775A41C110D)" and a "View your decryption key" button.

Symmetric vs Asymmetric Encryption



Asymmetric Cryptography

- **Pair of keys**

Encrypt with one key, decrypt with another key.

You can encrypt with **public** key and decrypt only with **private** key

You can encrypt with **private** key and decrypt only with **public** key

- **Public key** – share the key with anyone – [Public-Key Cryptography](#)
- **Private key** – do not share, keep private
- **Keyserver** – server that provides directory of OpenPGP keys

Thunderbird – easy way to start with OpenPGP

Thunderbird - <https://www.thunderbird.net/>

Mac:

```
brew install --cask thunderbird
```

Kali Linux (Thunderbird is probably installed already):

```
sudo apt update
```

```
sudo apt install thunderbird
```

Windows:

<https://www.thunderbird.net/> Click Download. Install

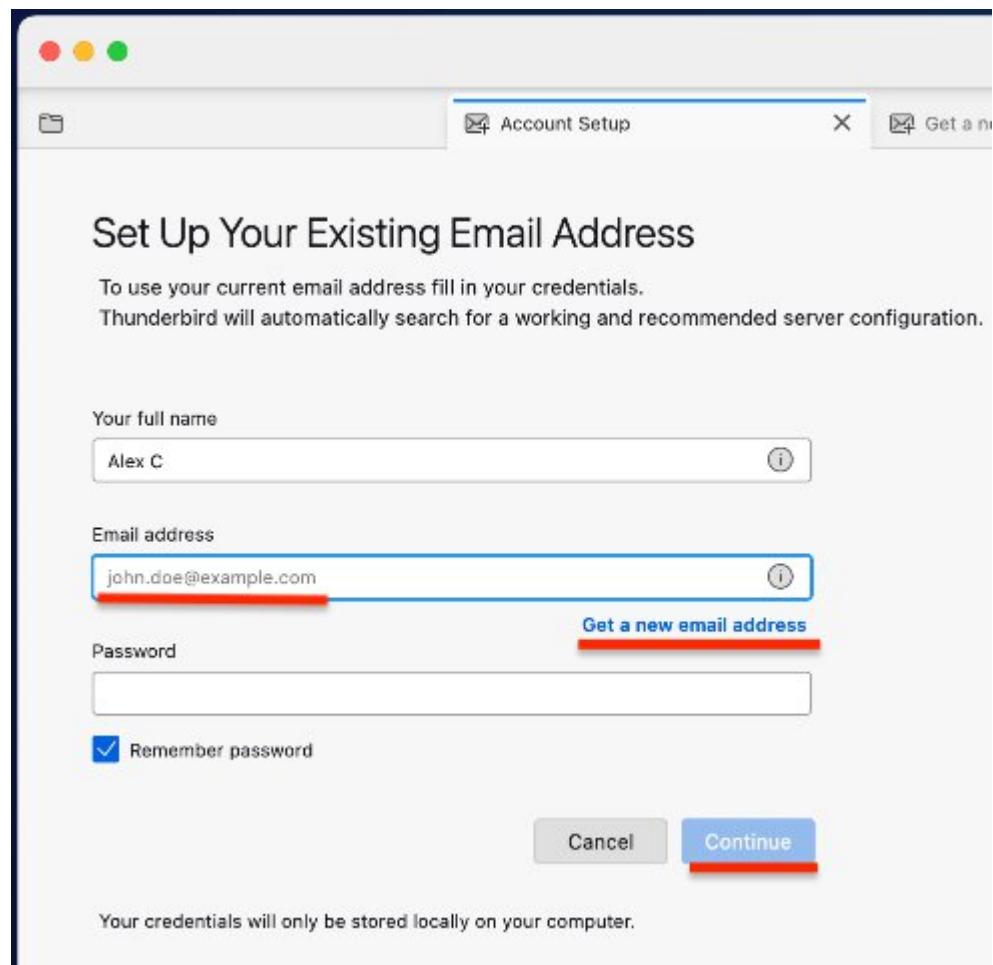
Thunderbird: Add Account

Sign in with your existing email

NOTE: For Google and Outlook
do not enter password, Click Continue

OR

Get a new email address



Thunderbird: Generate new Key Pair

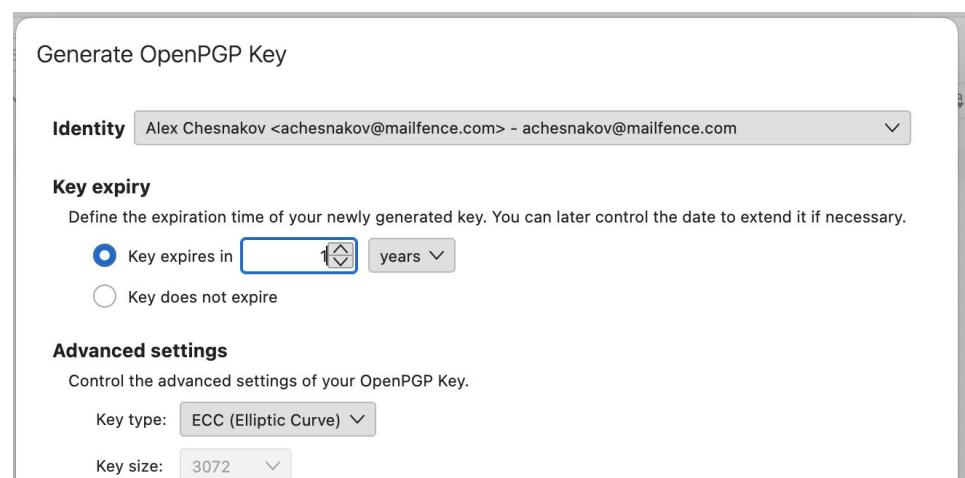
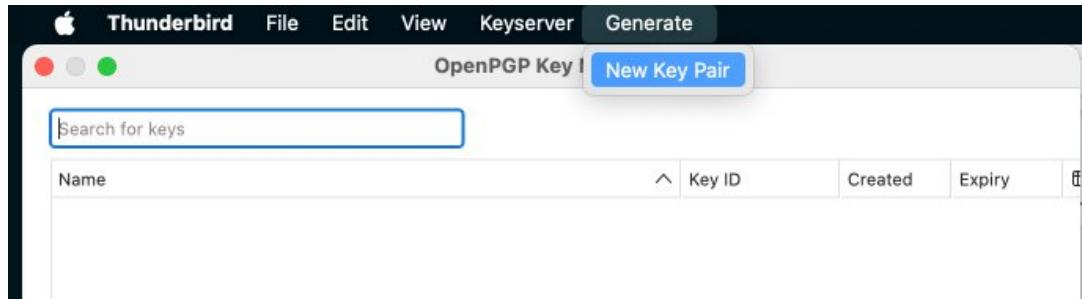
In Thunderbird:

Tools -> OpenPGP Key Manager

Generate -> New Key Pair

Defaults will work

You may Expire your first keys in 1 year
and use ECC Key (much shorter)

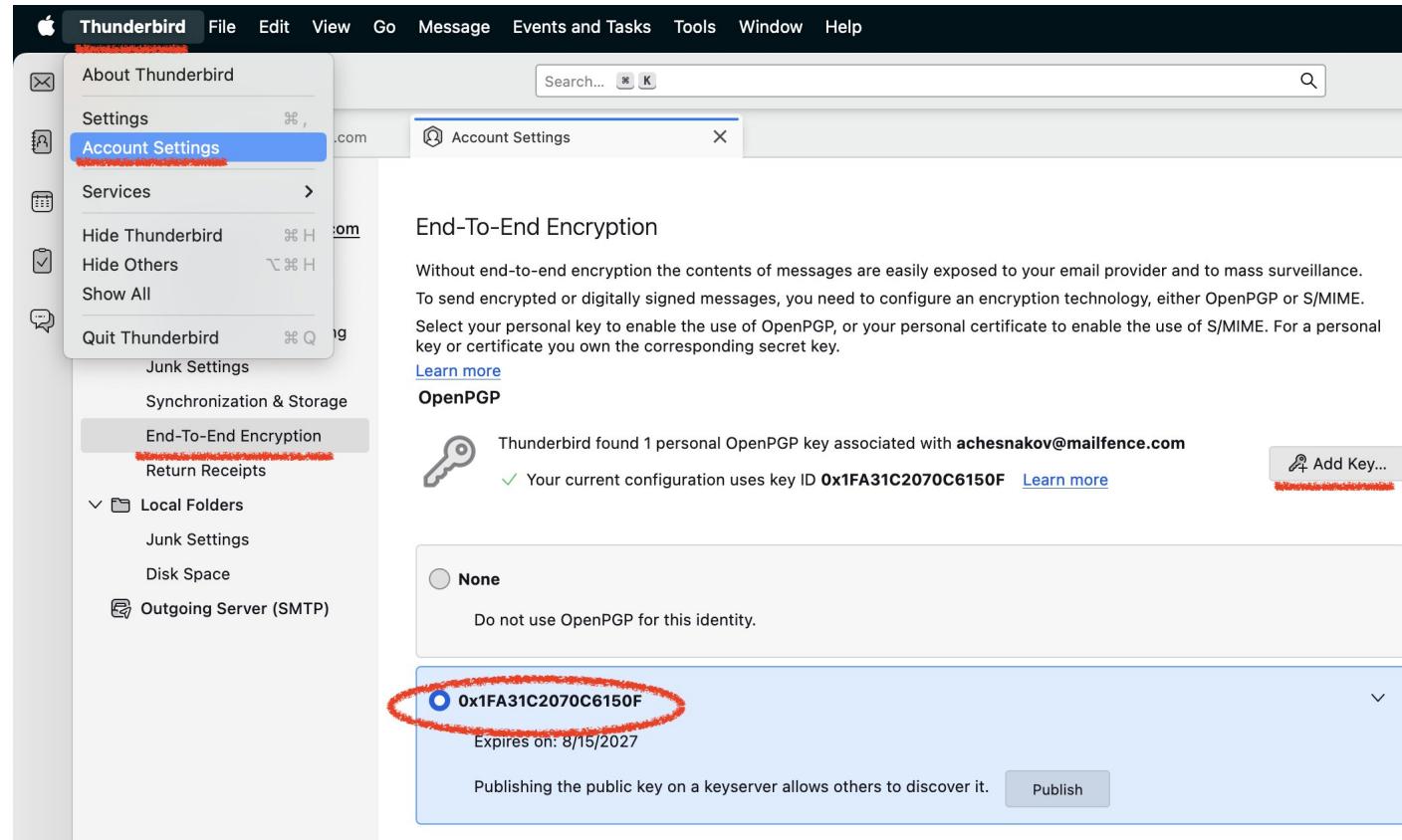


Thunderbird: use the OpenPGP Keys

In Thunderbird:

Thunderbird ->
Account Settings ->
End-To-End Encryption ->
Choose your Keys

May need to Add Key
(if keys do not shows up)

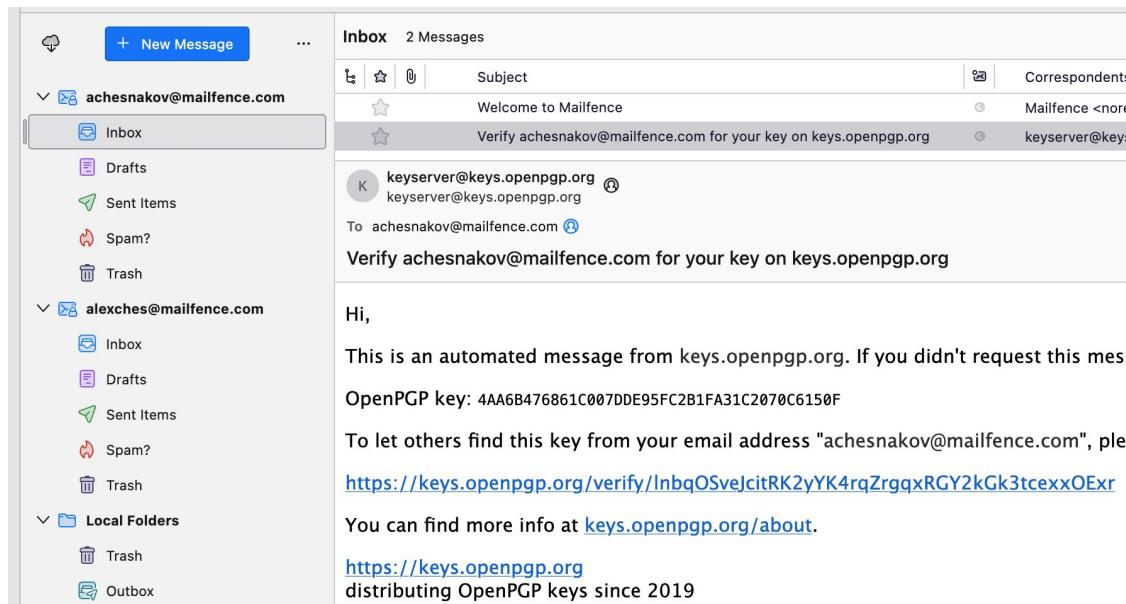
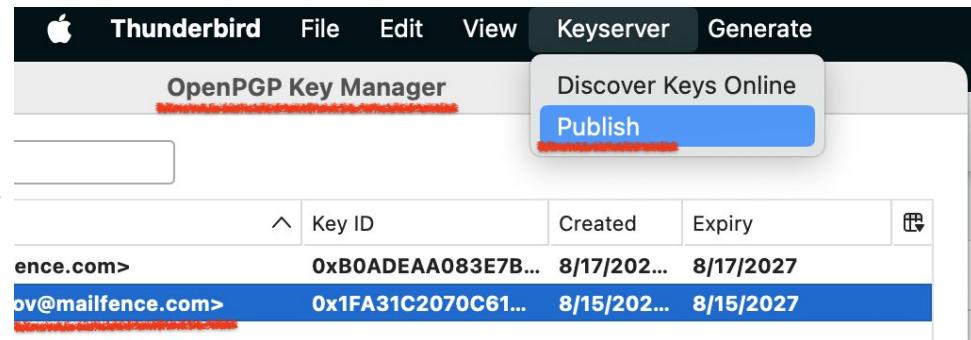


Publish Keys (Optional)

You may Publish the key
This will help others find your OpenPGP keys
but gives away info about you (privacy)

You should receive an email and
confirm that you the owner of
this email account

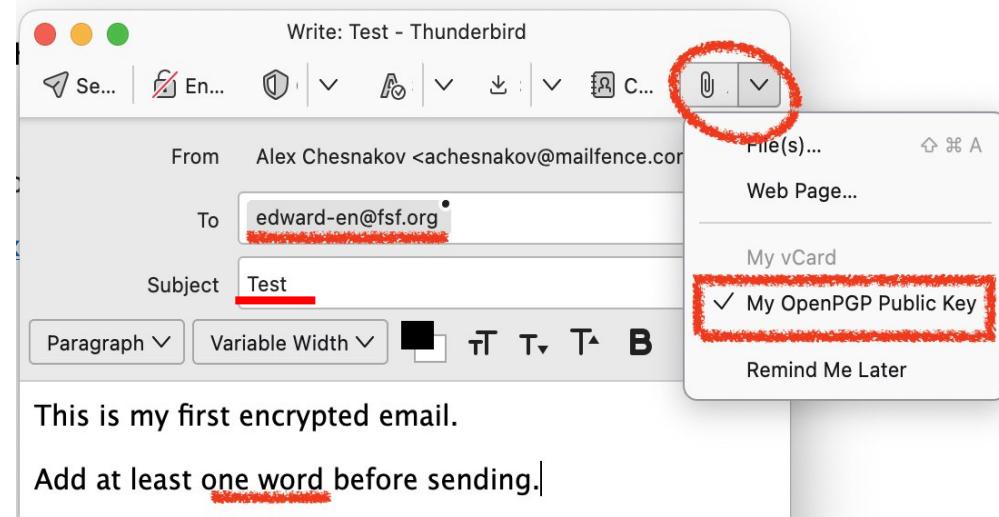
This will make the key discoverable
through the search at
<https://keys.openpgp.org/>



Email Edward your PubKey

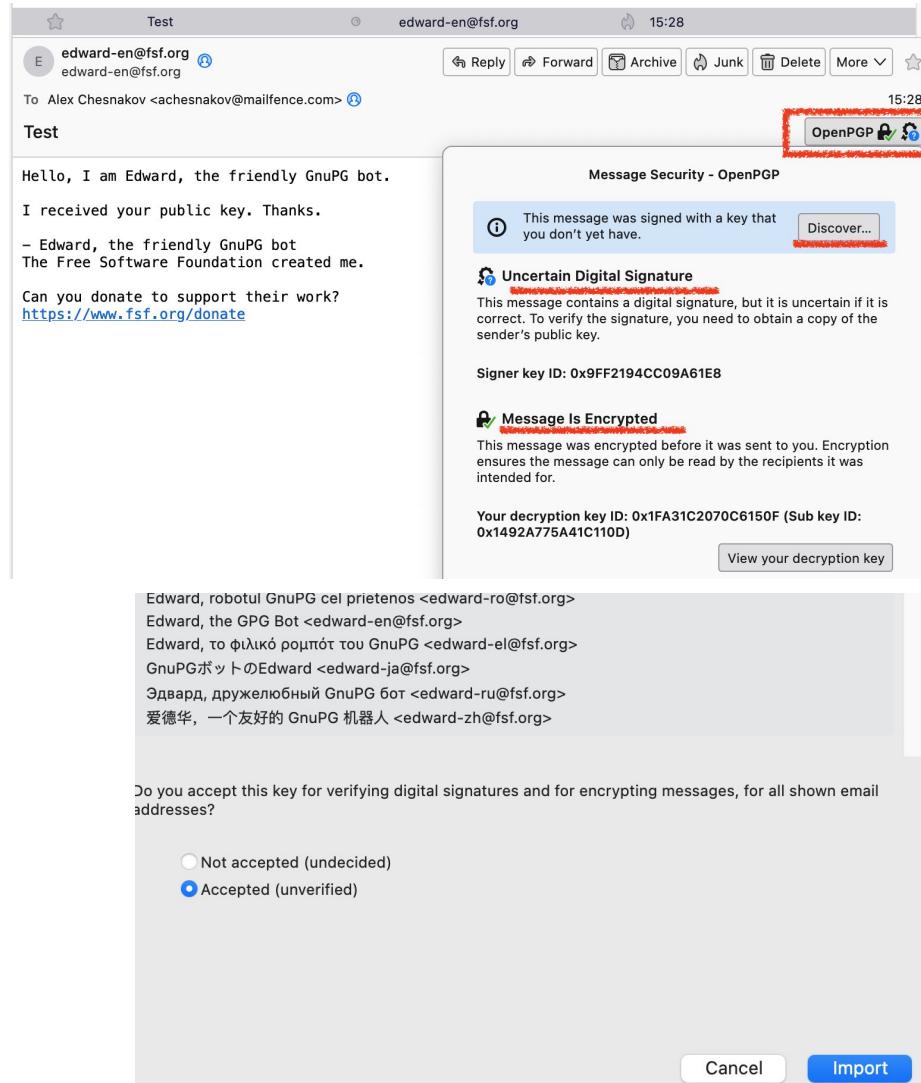
Free Software Foundation (FSF)
project to help start with OpenPGP

- Create email and attach Public Key
- Add one word to **Subject & Body**
- Sent to **edward-en@fsf.org**



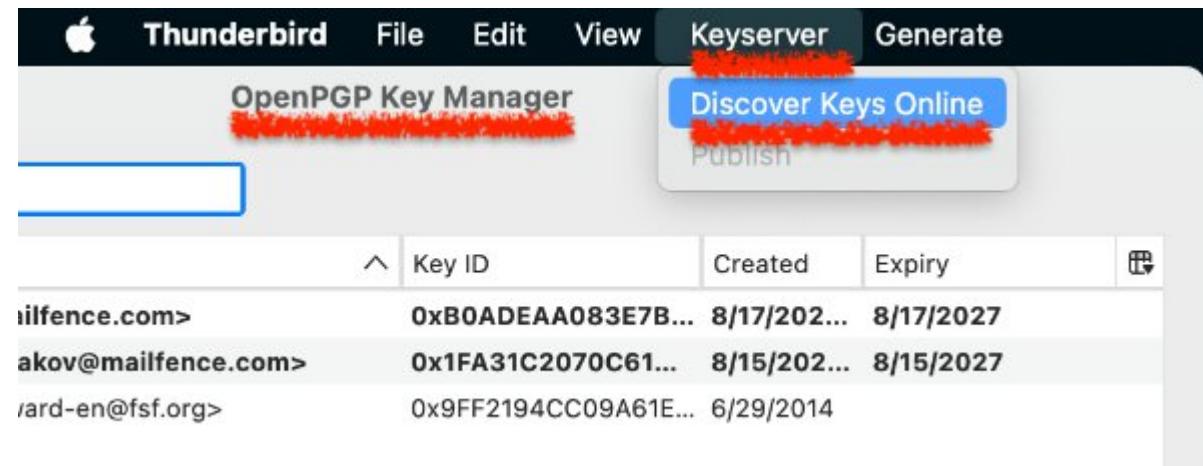
Receive First Encrypted Email

- Congrats, you received Encrypted Email
- Edward has our Public Key and can encrypt message with our PubKey
- We haven't added Edward's PubKey so we are not sure if Edward is Edward
- We can Click Discover
- We initiated the email, we can Import and choose Import Accepted



Discover PubKeys Online

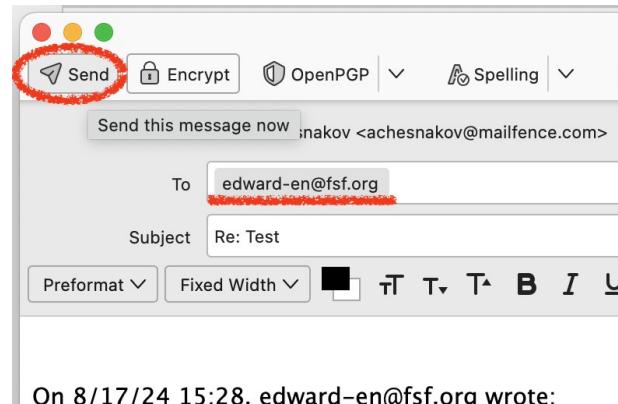
Thunderbird->
Tools->
OpenPGP Key Manager->
Keyserver->
Discover Keys Online



Try to find and Import
– edward-en@fsf.org
– achesnakov@mailfence.com

Send First Encrypted Email

- Reply to Edward and be sure to:
 - Encrypt
 - Digitally Sign
- Click Send
- Receive a response message
- Notice three dots instead of Subject



On 8/17/24 15:28. edward-en@fsf.org wrote:

A screenshot of the OpenPGP menu in a mail client. The "OpenPGP" menu is open, showing several options: "OpenPGP S/MIME", "Encrypt", "Encrypt Subject", "Digitally Sign", "Key Assistant", and "Key Manager". The "Encrypt" and "Encrypt Subject" options are circled in red. The background shows the message body with a response from "edward-en@fsf.org".

On 8/17/24 15:28, edward-en@fsf.org wrote:

Hello, I am Edward, the friendly GnuPG bot.

I received your public key. Thanks.

Receive an Encrypted Email from Edward

- Receive a response message
- Notice three dots instead of Subject

The screenshot shows an email inbox with four messages. The first message is from "edward-en@fsf.org" with subject "Test" sent on 8/17/24, 15:28. The second message is from "edward-en@fsf.org" with subject "..." sent on 8/17/24, 16:19. The third message is from "edward-en@fsf.org" with subject "..." sent on 8/17/24, 16:19. The fourth message is from "edward-en@fsf.org" with subject "..." sent on 8/17/24, 16:19. The message from 8/17/24, 16:19 is expanded, showing it was sent to "Alex Chesnakov <achesnakov@mailfence.com>". The message content is as follows:

Hello, I am Edward, the friendly GnuPG bot.
I received your message and decrypted it.
Your signature was verified.
I received your public key. Thanks.
Here's a copy of your message:

On 8/17/24 15:28, edward-en@fsf.org wrote:
Hello, I am Edward, the friendly GnuPG bot
I received your public key. Thanks.
- Edward, the friendly GnuPG bot
The Free Software Foundation created me.
Can you donate to support their work?
<https://www.fsf.org/donate>

The message is shown with a redacted subject and body. A tooltip for the message content indicates it is an OpenPGP message with a good digital signature and a decryption key ID of 0x1FA31C2070C6150F.

Check if email is encrypted

- Log into email provider web portal
- See how the message is looks like
- Two Files
 - encrypted.asc
 - PGP/MIME version identification
- Mailfence will offer to decrypt, but you have to upload your PRIVATE Key



https://mailfence.com/flatx/index.jsp#tool=mail&folderoid=456817342

New Refresh Reply Reply to all Send again Forward Delete Move ...

edward-en@fsf.org 16:19
Test edward-en@fsf.org 15:27

Alex Chesnakov <achesnakov@mailfence.com>
To: edward-en@fsf.org

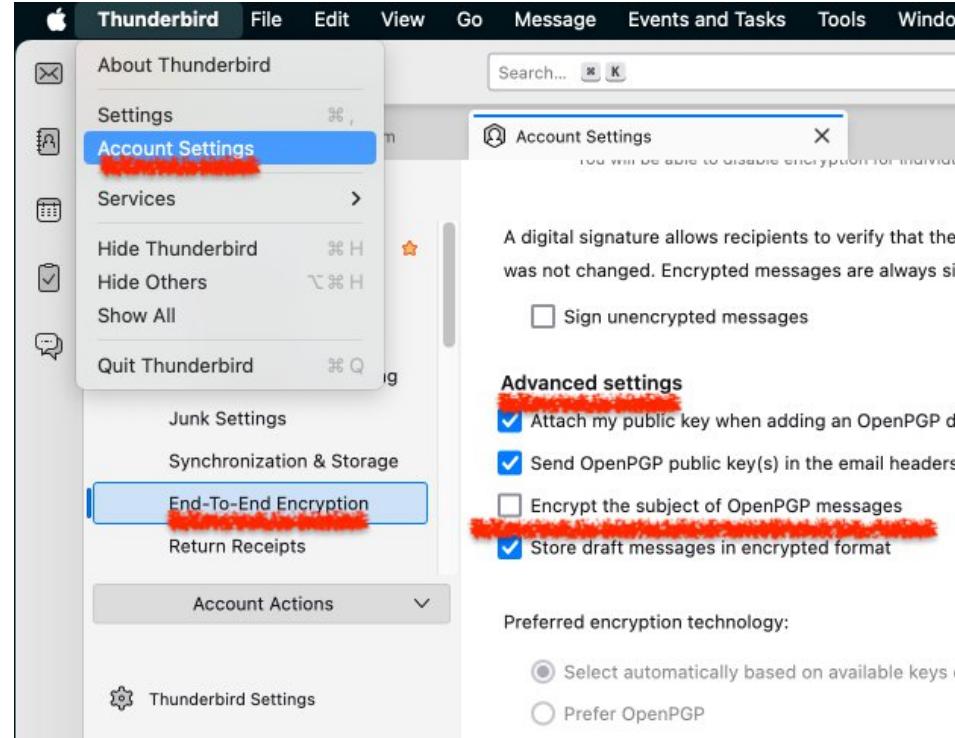
PGP/MIME version ... encrypted.asc

-----BEGIN PGP MESSAGE-----

wv4DFJKnadaQcEQ0SAQdAgmbUDEOQKHSovKIXAA7EuSKeC3l9WbWAqlYVv7jmwvOn3uzB0BIQwW
OeDP3zsvBdrRWDpmehc4eFTMZZYRkS0b1aFyf1sN3dy830lc9dwcbMAod322QFNLAQo5fc9k
OxZv947w1aWKM6fYcn9rD06bMTx8/9q3nrHVa+x+rtrhkDITMPDWf4YqRsok93007JdTSOMlsBq
Xhxy3Gg4pseUijYjZvmCbJY0mjMxx7k3N7+B4ja2nt3F8WKb1B2S1inSU3KH0dB/lnv3irjB
ApF9z/4+9m8/rEtLicoE5BpRh/+NdxfrHeRaHylI3xnXbc2gahBZYKMUuge+y7505MmLnjQdnd
F8Y84q7sXo7c4hjr4IJlwU+k8d76tBADGRFuQYjbztAQgi93wtSIVelib7pNI/R2uCF6V2DgD
SsscyZaz1AnPSXufri5V3FGrOCJYxdm/G9HlrgC3bdvBeGeGR/Qj+GfA17azhIiIoVTrQbPIPOsPr
vg5Ev6506Yh+ZW0zelQcfwLLgbT1H/lSmso7gkMpaJMou2OAH2PeVniXAw0akhVCPdnjlBz
CAHKLqTrZP50fMV3PGH/wuMqFCtZbODGbyVZVzys+Bx/WKBW/8ICXtQ+jUsJBmpewXVzRyPbn3+
xDbVOZmtphGdq50RrUg/XND1veW/SyGrq2NRVhOrxLeMmb06t7EZQfpCa3b0lh5LYFKDkqjY/+
VumPRYjE5d2GPB8/EpbBXQWb0shzLuHmlh6G8157H337tsPbQuLVqmmItmy3F
ZzaBU4me9gyugnTWOjSUA2l/Nl/V2c0aaPyhob0a4oskhYXzZPLmD3znAcnwfr+rEVdl7E40
pYp+whax8cp82Ujh55aaR49PnlbYz36/HesDPTDgVtUvju9QpBtRpIplYfj9asPV0Y1FO7
OMQ9sWEoZjofr2ucqwPmAm+ZhvwHMZ+UqPvdZ/0P8eVzWtdeEnUroii2OH2PeVniXAw0akhVCPdnjlBz
qE35b3GfdVlNgJGZza375N087wjC+NeOLqmba/Zpt/bkLi/pixYIIn9bW6LqfbT6m2CAP5
ZckC9kWzHpmZv3ugWJHjkNqLmtysi9hRc5SD16e/Tj5k6mLH5YWtbZ0FybzRHH1kX
m7Ozhwzle4kgQcXbVBNKLXC6peue7VfJDTEGD6VC4mf75o4SPEnjd1RHAC+3OnbcXamoro/
9835dbObD15Wq4AkP0c25T80pe/zFEKKaG1NXX/Q5r+kKcrNn+4+znkWgzeB77H+ujmQ/q79
ABVftKoqVYePMOJ6lU+dYRMILcyOkoxWTvhArlx4Ar/BsMjznJCLZ/rIaRiSj0VCLlnxsVAUxVz
Eahx27Aow-TkURlvzsRa1AuWstV5A1ZROU4A+j9izrn7pchtWzeza4wi0lPfQrCjQj0l
Ds1phm2MRcis8+nL5T5k++BYNNNGk/EC9g/brDprJoNoMqMd+D5Dpk+oYt37MsK9jdU1755uC2
T/SE6bnnmYFrd80/4e/HcfMr1mG5QqcJnC1FqFSHs76On1NugVu3BQIBm9t7p0WuJWQmthYirGct9E4t
ShYtQZ-kbthx/BM1c5C9Ly8G7O/jxt65yAC2W/Mpk9m/na7zaaw6wvPwpWQmthYirGct9E4t
E5lgBS6druQcMpf1cxh3YVkk47sdsf07VjExK21KzYdTgtmsCtbev7pch71ML01P1
ORXkOT4Q2SSWW4WF4m+/emHaMIHj9135sGXiPj4dVnNWFtC+9BAXNep76yBv0e+71ML01P1
HVzTEq5A+4zzk2q64yw7RQfUHRe1C2dv4KxWUtwmVbdR4i4u2XmuSVfjbvDzylh220pTz30IM+A
XGPYnYlrd+9ly8u+1Zp5Yn1fQKrnkn01ZrBja9+YK3Pbfa0GJSwu+9NPLfOsikL05'/+nbJj6
iyeH8GVs3pjR19tHAZEIOGG2ZVYy5OZEUal0xu2VdLOVOpQsQYGYu+E4j3P/r1m2nJmUePvDn
RstrYfu/nxfkj5ihZ5CuWvy1v6kQH2WeQsUsQpEPBXy+OvBwIWQdCommxr/7748B09QvqAbv
DkWraRj90W0O1pcDJANRjWMJ3UQdtCbu+2BegeYNmhvJrzkkhy+jA4cZOAOHNHjUZB17QaWq//
4LZizGPRHk8w4EWt5k12ZhmFHETuR7f2gtD52RGGCemUge/E/m3yK1DzCkYADA1/79U146R
7wVKWVpnk1E9UxrWzj0cYNH3xw+qtH9QGlus01RpYH43tCeB6c1QZr8rN35m/8Bz0EKF3+ZL
YzYuqcX1vm5s5ar9YK3PbQvTZ4ldtYaApBpXw+YkslyQWlqv+80v7tWhTrOvxAZZ/ovMMtoh5
isTcQFH0gwdsMcTv2B6We8goLXEpfqvwe3Q1VfLw60KpCenglplQUBdTp5xSkUlx8dgrRenaGr
XHtCqfjUN5Pr7ThPn23LrkVQHcjrsrukzg18QoAcEhukuPa44kYh+AmgQhA7o0o7f71RqVpUgP
5imt5HGeF59hmMetrlQL9rmFrxLSLGNsWfUYSSsfAoYBhuOPBnx5HwrmnCkyCa5lOrldLVd/
NYP6L06uSAURMmqgAyPTCQLUWYMF4WRKDtibzuXBNjP501ZLnenbmO2v2/74Kedsa1RHYqjv9
naNTdoiAcwc07PnbtYQH/84LZZZ9XcfzKOSifruAuQdp0SfypdilXfmlkjHGoxv91NwUeYca550=
FdJi3u6EKs/mfGu/jbgvp2DHewVz1cmAb1CqfTkH10fRECI7tuqp0o4x0wz17boej0kL75
h64D/WRFJgaMu/HYUta6x6byoyLt3Cd/YqDOpp9nScVcl+kTjWEv7mr7QU8T2jF4j0YRfqA9G
LOLOuDx2R9dk7n/4oY6hwHrDevipKnp8GATrN0vb5NT5FvIJS2g2b0mcISRUefgK0lmiq-q/k/
ZV018aLlkmoSpfRkyle2eHOADctadSNCHGqqPwfRip3ztyCW69dKqmMDxD8dkAHlCv5v/eA8ylK
5MzAqCAnL9TgmddPzSysqDwVc9csCsGute32AiAbmKFM57nb6776oo7f7oYRawPBxsk110Y
8K3tuUOy/a3vMAwHGDeqkCICjyxt+Xx7j0oH7ASU1ZKGr17PkC8Ukr7R0mKjt0RyCe550=
=ATIX
-----END PGP MESSAGE-----

Subject Encryption

- OpenPGP do not offer Encryption for Subject
 - Thunderbird encrypts Subject by default
 - They use non-standard implementation
 - Works fine between Thunderbird clients
 - Do not work between everything else
 - Choose your subject ("Hello", "Test")
-
- **Do not encrypt subject**
 - Go to Account Settings->
End-To-End Encryption->
Advanced Settings->
 - Uncheck Encrypt Subject

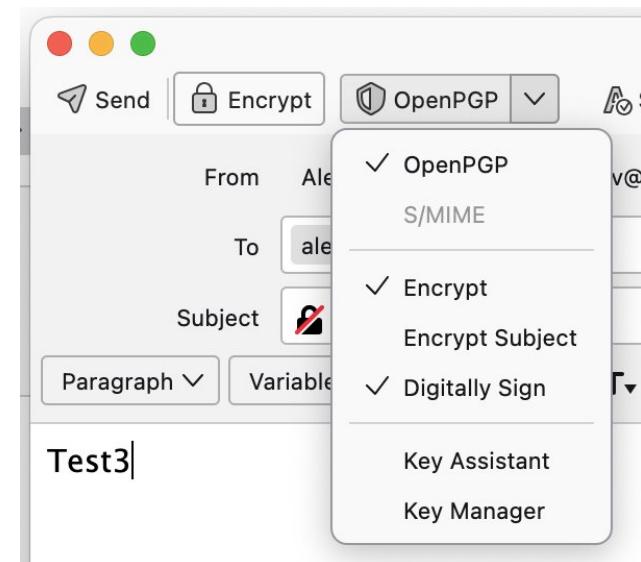


Exchange encrypted email with a person

- Ask somebody around for their email that has OpenPGP keys
- As a backup, you can send email to me - achesnakov@mailfence.com
- Search for their keys at Thunderbird's OpenPGP Key Manager or at <https://keys.openpgp.org/>
- Import keys
- Click on the keys and Accept keys with "Yes, I've verified in person..."
- Create a new email to this person
- Be sure you Encrypt and Digitally sign the message

Other person:

- Receive the email
- Import Keys
- Click Reply to the email
- Encrypt and Digitally Sign the email before Sending



Advanced use of GnuPG (GPG)



The screenshot shows the GPGTools website. At the top left is the GPGTools logo (a blue padlock icon) and the text "GPGTools". At the top right is a "Support Plan" button. Below the header is a large section titled "GPG Suite" in bold black font. Underneath it, a paragraph reads: "One simple package with everything you need, to protect your emails and files." A red "Download" button is centered below this text, with the subtext "for macOS 10.15 - 13.x" underneath it. At the bottom of the main content area is a small note: "By downloading, you agree to our [Terms of Distribution](#)".

GPG Suite

One simple package
with everything you need,
to protect your emails and files.

Download

for macOS 10.15 - 13.x

By downloading, you agree to our [Terms of Distribution](#)

Includes a 30-day trial of GPG Mail. For continued use of GPG Mail, please purchase a [support plan](#)

Gpg4win Components

Gpg4win is an installer for Windows and contains several Free Software components:



GnuPG

The backend; this is the actual encryption tool.



Kleopatra

A certificate manager for OpenPGP and X.509 (S/MIME) and common crypto dialogs.



GpgOL

An add-in for Microsoft Outlook providing email cryptography. Supports both IMAP/SMTP and MS Exchange servers.



GpgEX

A plugin for Microsoft Explorer (file encryption).

Install GnuPG (CLI) and GUI tools

GnuPG - <https://gnupg.org/>

Mac. GPG Suite. Includes CLI and GUI:

```
brew install --cask gpg-suite
```

Kali Linux. GnuPG and Kleopatra (gpg is pre-installed):

```
sudo apt update
```

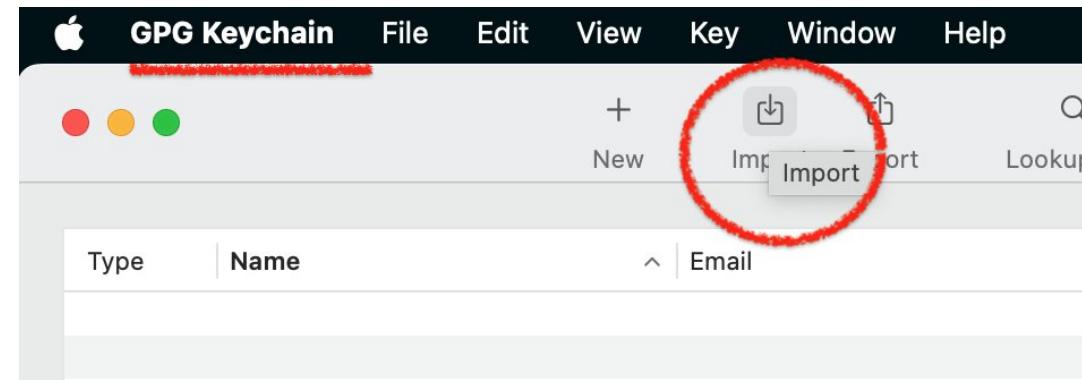
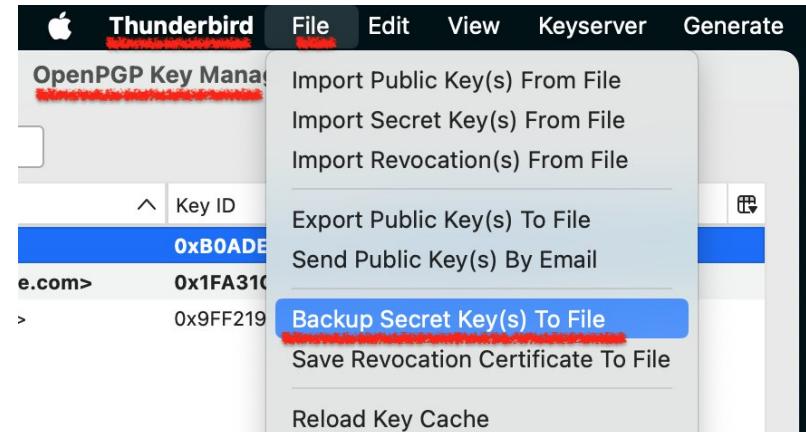
```
sudo apt install gpg kleopatra
```

Windows. Gpg4win. Includes GnuPG and Kleopatra:

<https://gpg4win.org/> Click Download. Install

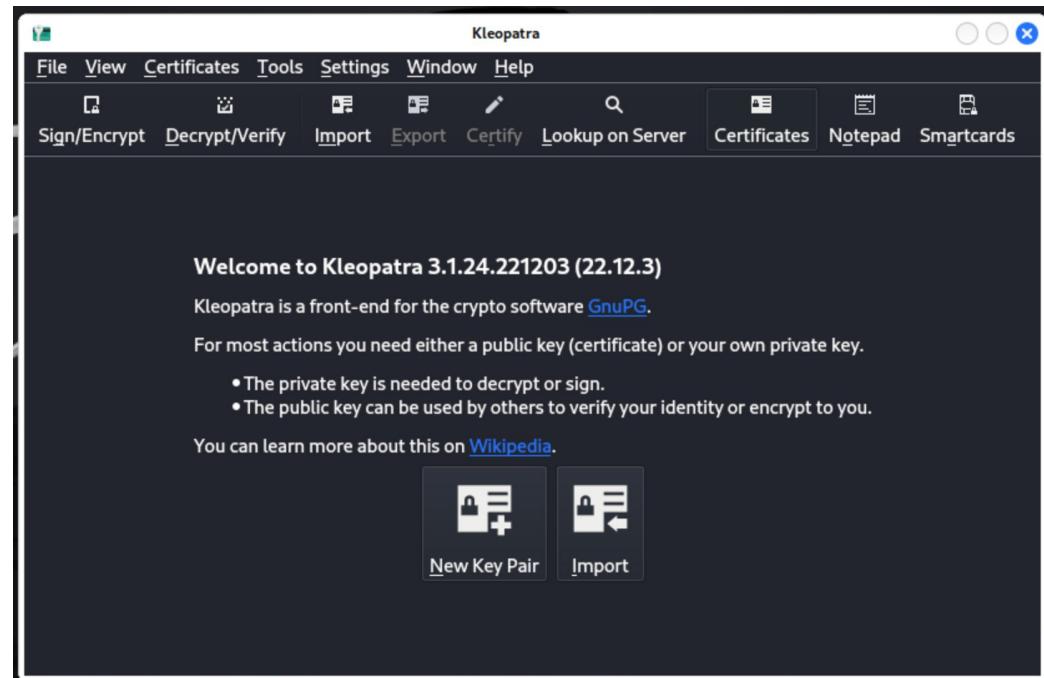
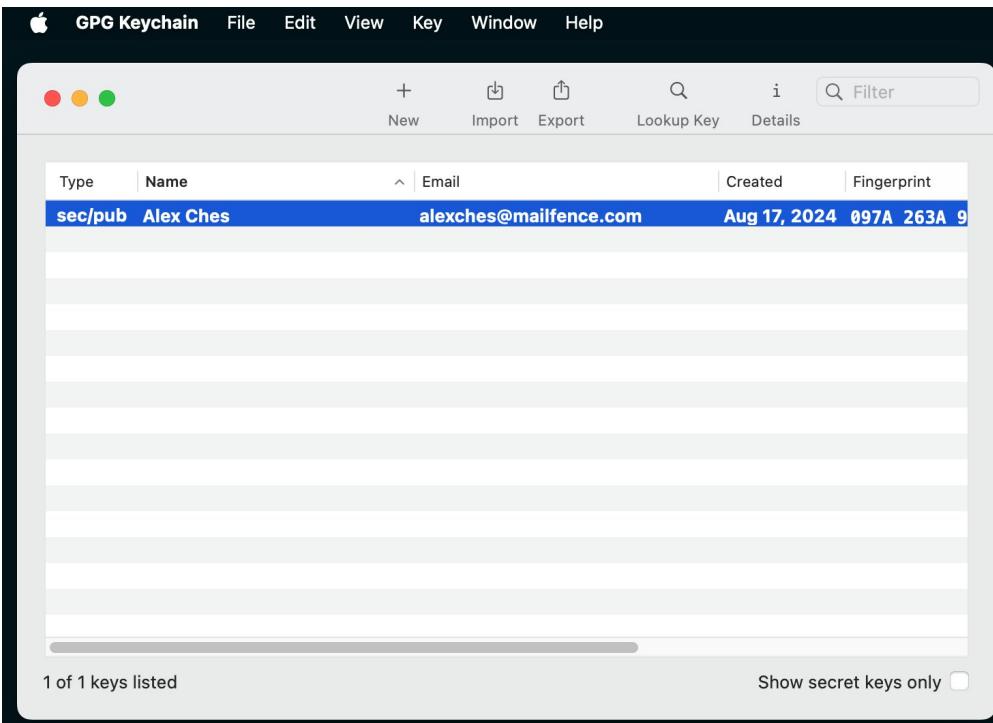
Export Keys from Thunderbird to GnuPG

- Thunderbird can't sync keys with GnuPG
 - Export Secret (Private) Keys from Thunderbird
 - Thunderbird will export Private and Pub Keys
 - Choose Password for Private (Secret) Keys
-
- Import keys to GPG Keychain or Kleopatra



CLI vs GUI

- CLI version offers full functionality
- For simple use both CLI and GUI are good
- CLI version can be used in scripts and automation



List Keys

- **gpg --list-keys** – List Public Keys
- **gpg --list-secret-keys** – List Private Keys

```
› 20:02 alexc@Alexs-MBP /Users/alexc
% gpg --list-keys
/Users/alexc/.gnupg/pubring.kbx
-----
pub ed25519 2024-08-17 [SC] [expires: 2027-08-17]
097A263A90BF7916CFC41014B0ADEAA083E7BE88
uid      [ultimate] Alex Ches <alexches@mailfence.com>
sub cv25519 2024-08-17 [E] [expires: 2027-08-17]

pub ed25519 2024-08-16 [SC] [expires: 2027-08-16]
4AA6B476861C007DDE95FC2B1FA31C2070C6150F
uid      [ unknown] Alex Chesnakov <achesnakov@mailfence.com>
sub cv25519 2024-08-16 [E] [expires: 2027-08-16]

pub rsa2048 2014-06-29 [SC]
F357AA1A5B1FA42CFD9FE52A9FF2194CC09A61E8
uid      [ unknown] Edward, the GPG Bot <edward-en@fsf.org>
uid      [ unknown] Edward the GPG Bot <edward@fsf.org>
uid      [ unknown] GnuPGボットのEdward <edward-ja@fsf.org>
uid      [ unknown] Edward, l'amichevole bot GnuPG <edward-it@fsf.org>
uid      [ unknown] Edward, le gentil robot de GnuPG <edward-fr@fsf.org>
uid      [ unknown] Edward, el simpático robot GnuPG <edward-es@fsf.org>
uid      [ unknown] Edward, o amigo robô de GnuPG <edward-pt-br@fsf.org>
uid      [ unknown] Edward, robotul GnuPG cel prietenos <edward-ro@fsf.org>
uid      [ unknown] Edward, arkadas canlısı GnuPG botu <edward-tr@fsf.org>
uid      [ unknown] Edward, der freundliche GnuPG Roboter <edward-de@fsf.org>
uid      [ unknown] Эдвард, дружелюбный GnuPG бот <edward-ru@fsf.org>
uid      [ unknown] Edward, το φιλικό ρομπότ του GnuPG <edward-el@fsf.org>
uid      [ unknown] 爱德华, 一个友好的 GnuPG 机器人 <edward-zh@fsf.org>
sub rsa2048 2014-06-29 [E]
```

```
› 20:03 alexc@Alexs-MBP /Users/alexc
% gpg --list-secret-keys
/Users/alexc/.gnupg/pubring.kbx
-----
sec   ed25519 2024-08-17 [SC] [expires: 2027-08-17]
097A263A90BF7916CFC41014B0ADEAA083E7BE88
uid      [ultimate] Alex Ches <alexches@mailfence.com>
ssb   cv25519 2024-08-17 [E] [expires: 2027-08-17]

sec   ed25519 2024-08-16 [SC] [expires: 2027-08-16]
4AA6B476861C007DDE95FC2B1FA31C2070C6150F
uid      [ unknown] Alex Chesnakov <achesnakov@mailfence.com>
sub cv25519 2024-08-16 [E] [expires: 2027-08-16]
```

Export Keys

```
gpg --export-secret-keys --armor [keyID] > my_pgpkey.private.asc  
gpg --export --armor [keyID] > my_pgpkey.public.asc
```

[keyID] – **Last 8 or 16 digits of Fingerprint**
may use Email Address
or Full Fingerprint

--armor – Export as ASCII text

```
>20:36 alexc@Alexs-MBP /Users/alexc  
% gpg --list-keys edward-en@fsf.org  
pub rsa2048 2014-06-29 [SC]  
[F357AA1A5B1FA42CFD9FE52A9FF2194CC09A61E8 ]  
uid [ unknown] Edward, the GPG Bot <edward-en@fsf.org>  
uid [ unknown] Edward the GPG Bot <edward@fsf.org>  
uid [ unknown] GnuPGボットのEdward <edward-ja@fsf.org>  
uid [ unknown] Edward, l'amichevole bot GnuPG <edward-it@fsf.org>  
uid [ unknown] Edward, le gentil robot de GnuPG <edward-fr@fsf.org>  
uid [ unknown] Edward, el simpático robot GnuPG <edward-es@fsf.org>  
uid [ unknown] Edward, o amigo robô de GnuPG <edward-pt-br@fsf.org>  
uid [ unknown] Edward, robotul GnuPG cel prietenos <edward-ro@fsf.org>  
uid [ unknown] Edward, arkadaş canlısı GnuPG botu <edward-tr@fsf.org>  
uid [ unknown] Edward, der freundliche GnuPG Roboter <edward-de@fsf.org>  
uid [ unknown] Эдвард, дружелюбный GnuPG бот <edward-ru@fsf.org>  
uid [ unknown] Edward, το φιλικό ρομπότ του GnuPG <edward-el@fsf.org>  
uid [ unknown] 爱德华, 一个友好的 GnuPG 机器人 <edward-zh@fsf.org>  
sub rsa2048 2014-06-29 [E]  
  
>20:36 alexc@Alexs-MBP /Users/alexc  
% gpg --export --armor C09A61E8 > edward-en_fsf.org.public.asc  
>20:36 alexc@Alexs-MBP /Users/alexc  
% gpg --export --armor edward-en@fsf.org > edward-en_fsf.org.public.asc  
>20:37 alexc@Alexs-MBP /Users/alexc  
% gpg --export --armor F357AA1A5B1FA42CFD9FE52A9FF2194CC09A61E8 > edward-en_fsf.org.public.asc  
>20:38 alexc@Alexs-MBP /Users/alexc  
% cat edward-en_fsf.org.public.asc  
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mQENBF0wfzoBCADpwK6sGC3EUzgD7IW1X5ZDR1nC5/rcXacAJLarPpvQBEz4pwjT  
joAzATM7F9RwIzJ3hJTZhYqY4cfiGLKSrnd8GPC8A4QkxXIaQ0hLpcSBsbtZJp  
o2i0zL2fRHmW2ZlnSHXPKbDwx5p0NcdQfjL9i2Yo31aLI0/ChhnSuyvIzn0jaCSC  
/06x2C4m81Lu+B4UTDpl8y6ChtphUxyFGd7RXDXmkYQrxVqJbXKuSVmNMhM09myG  
7i0110Yl0cCxv3TYDn0k+a49BbMGB9w14eDTE86HEzP.iM+dhEnhTOW/+1N4Yl0UV
```

Delete Keys

```
gpg --delete-secret-keys [keyID]
gpg --delete-keys [keyID]
```

[keyID] – **Last 8 or 16 digits of Fingerprint**
may use Email Address
or Full Fingerprint

```
▶20:49 alexc@Alexs-MBP /Users/alexc
% gpg --delete-keys edward-en@fsf.org
gpg (GnuPG/MacGPG2) 2.2.41; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub rsa2048/9FF2194CC09A61E8 2014-06-29 Edward, the GPG Bot <edward-en@fsf.org>

Delete this key from the keyring? (y/N) y
▶20:50 alexc@Alexs-MBP /Users/alexc
% gpg --list-keys
/Users/alexc/.gnupg/pubring.kbx
-----
pub ed25519 2024-08-17 [SC] [expires: 2027-08-17]
097A263A90BF7916CFC41014B0ADEAA083E7BE88
uid [ultimate] Alex Ches <alexches@mailfence.com>
sub cv25519 2024-08-17 [E] [expires: 2027-08-17]

pub ed25519 2024-08-16 [SC] [expires: 2027-08-16]
4AA6B476861C007DDE95FC2B1FA31C2070C6150F
uid [ unknown] Alex Chesnakov <achesnakov@mailfence.com>
sub cv25519 2024-08-16 [E] [expires: 2027-08-16]

▶20:50 alexc@Alexs-MBP /Users/alexc
%
```

Import Keys

```
gpg --import my_pgpkey.private.asc  
gpg --import my_pgpkey.public.asc
```

```
▶20:59 alexc@Alexs-MBP /Users/alexc  
% ls -lah ed*  
-rw-r--r--@ 1 alexc  staff  8.0K Aug 18 20:38 edward-en_fsf.org.public.asc  
▶20:59 alexc@Alexs-MBP /Users/alexc  
% gpg --import edward-en_fsf.org.public.asc  
gpg: key 9FF2194CC09A61E8: public key "Edward, the GPG Bot <edward-en@fsf.org>" imported  
gpg: Total number processed: 1  
gpg:                      imported: 1  
▶20:59 alexc@Alexs-MBP /Users/alexc  
%
```

Keyservers

```
gpg --keyserver htps://keys.openpgp.org --search edward-en@fsf.org
```

We can **skip --keyserver htps://keys.openpgp.org**
if we are using **default** keys.openpgp.org server,
not custom, like pgp.mit.edu

```
gpg --search achesnakov@mailfence.com
```

Email address may have multiple Keys,
need to specify specific Key
gpg --recv-keys 1FA31C2070C6150F

```
gpg --send-keys 1FA31C2070C6150F
```

```
>21:13 alexc@Alexs-MBP /Users/alexc
% gpg --search achesnakov@mailfence.com
gpg: data source: https://keys.openpgp.org:443
(1) Alex Chesnakov <achesnakov@mailfence.com>
      256 bit EDDSA key 1FA31C2070C6150F, created: 2024-08-16
      Keys 1-1 of 1 for "achesnakov@mailfence.com". Enter number(s), N(ext), or Q(uit) > n
>21:14 alexc@Alexs-MBP /Users/alexc
% gpg --recv-keys achesnakov@mailfence.com
gpg: "achesnakov@mailfence.com" not a key ID: skipping
>21:14 alexc@Alexs-MBP /Users/alexc
% gpg --recv-keys 1FA31C2070C6150F
gpg: key 1FA31C2070C6150F: "Alex Chesnakov <achesnakov@mailfence.com>" not changed
gpg: Total number processed: 1
gpg: unchanged: 1
>21:14 alexc@Alexs-MBP /Users/alexc
% gpg --send-keys 1FA31C2070C6150F
gpg: sending key 1FA31C2070C6150F to htps://keys.openpgp.org
>21:15 alexc@Alexs-MBP /Users/alexc
%
```

Generate Keys

```
gpg --full-generate-key
```

Revocation Certificate:

- Save it in a safe place
- Upload to keyserve if you lost your Private Key
- Decryption will still work after you upload Rev.Cert.
- Somebody need to search keyserver to notice that the key was revoked

```
21:46 alexc@Alexs-MBP /Users/alexc
% gpg --full-generate-key
gpg (GnuPG/MacGPG2) 2.2.41; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072)
Requested keysize is 3072 bits
Please specify how long the key should be valid.
 0 = key does not expire
 <n> = key expires in n days
 <n>w = key expires in n weeks
 <n>m = key expires in n months
 <n>y = key expires in n years
Key is valid for? (0) 1y
Key expires at Tue Aug 19 21:46:58 2025 PDT
Is this correct? (y/N) y

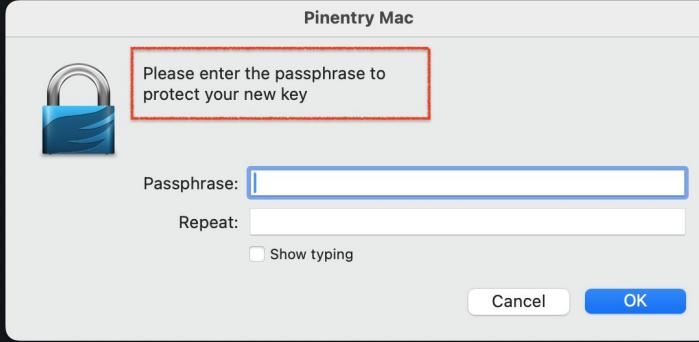
GnuPG needs to construct a user ID to identify your key.

Real name: Alex Ches
Email address: alexches@mailfence.com
Comment:
You selected this USER-ID:
  "Alex Ches <alexches@mailfence.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: directory '/Users/alexc/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/Users/alexc/.gnupg/openpgp-revocs.d/6FADBE92358625468E0D9CE6125468E82CF4320.rev'
public and secret key created and signed.

pub rsa3072 2024-08-20 [SC] [expires: 2025-08-20]
6FADBE92358625468E0D9CE6125468E82CF4320
uid Alex Ches <alexches@mailfence.com>
sub rsa3072 2024-08-20 [E] [expires: 2025-08-20]

21:48 alexc@Alexs-MBP /Users/alexc
%
```



Sign the message / file

```
gpg --sign file
```

file.gpg (input file + signature, binary)

```
gpg --clear-sign file
```

file.asc (original message + ascii signature)

```
gpg --detach-sign file
```

file.sig (separate signature, binary)

```
gpg --armor --local-user 70C6150F --detach-sig file
```

file.asc (specify private key, separate signature, ascii)

```
% gpg --armor --local-user 70C6150F --detach-sig my_message.txt
```

```
>23:10 alexc@Alexs-MBP /Users/alexc
```

```
% cat my_message.txt.asc
```

```
-----BEGIN PGP SIGNATURE-----
```

```
iHUEABYIAB0WIQRKprR2hhwAfd6V/CsfoxwgCMYVDwUCZsQzTAKCRAFoxwgCMYV  
D8iaAQD7x8ufLH93ZFercftj8eq1cHQSzmnFQ7nxw50csExFaQD9F2gmXDpLQLHF  
aVhQQw02k6xVzy6xcgrxDv1IRkNqCgE=
```

```
=XdoA
```

```
-----END PGP SIGNATURE-----
```

```
% cat my_message.txt
```

This is my message.

```
>22:31 alexc@Alexs-MBP /Users/alexc
```

```
% gpg --sign my_message.txt
```

```
>22:31 alexc@Alexs-MBP /Users/alexc
```

```
% cat my_message.txt.gpg
```

```
-----!a0000u000H0=00M-.NL00+0(I;000Y0D00  
000R1Y1E0*50 0+000000000)i~000sq00j000b0g60  
000Wr0M<0EcCX0f3#â<j5  
`0p060U0010%6~00~00g/3%
```

```
% gpg --clear-sign my_message.txt
```

```
>22:33 alexc@Alexs-MBP /Users/alexc
```

```
% cat my_message.txt.asc
```

```
-----BEGIN PGP SIGNED MESSAGE-----
```

Hash: SHA256

This is my message.

```
-----BEGIN PGP SIGNATURE-----
```

```
iHUEARYIAB0WIQQJeIY6kL95Fs/EEBSwreqgg+e+iAUCZsQqugAKCRCwreqgg+e+  
iFdSAQCSMcDIIXEUYCYAmK16/mS/HzKNiXP+pSZKPUKSY1SnyQD9G/W88/STb00d  
8qU94jTwxsMtkr0xHYYyvpqiNw6AHwU=  
=NyDr  
-----END PGP SIGNATURE-----
```

```
% gpg --detach-sign my_message.txt
```

```
>22:48 alexc@Alexs-MBP /Users/alexc
```

```
% cat my_message.txt.sig
```

```
-----
```

```
-----
```

Verify Signature

```
gpg --verify my_message.txt.sig
```

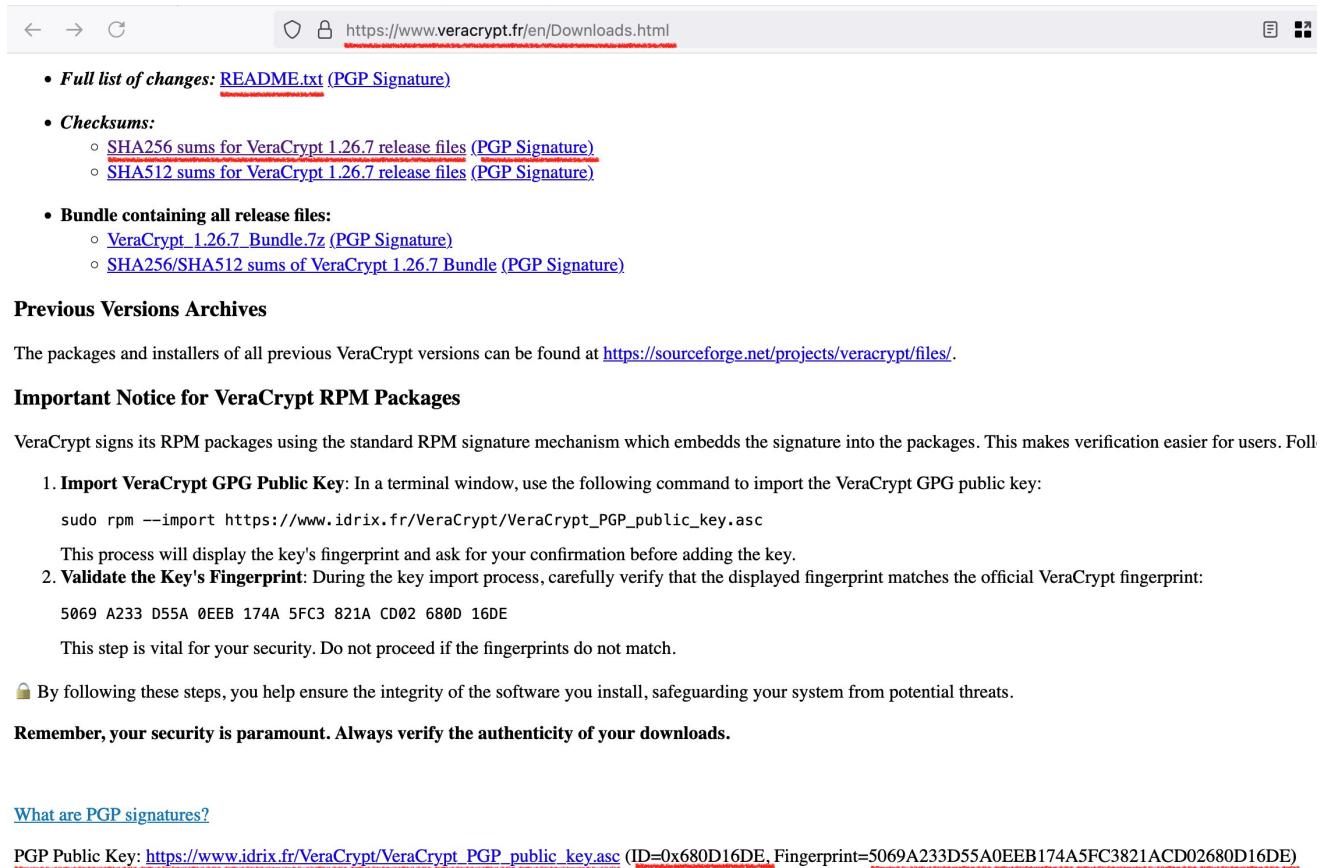
```
% gpg --verify my_message.txt.sig
gpg: assuming signed data in 'my_message.txt'
gpg: Signature made Mon Aug 19 22:48:39 2024 PDT
gpg:           using EDDSA key 097A263A90BF7916CFC41014B0ADEAA083E7BE88
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:  2  signed:  0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2025-08-20
gpg: Good signature from "Alex Ches <alexches@mailfence.com>" [ultimate]
```

Verifying Files from the Internet

Go to <https://www.veracrypt.fr/en/Downloads.html>

Download

- README.TXT
- veracrypt-1.26.7-sha256sum.txt
- veracrypt-1.26.7-sha256sum.txt.sig



The screenshot shows a web browser window with the URL <https://www.veracrypt.fr/en/Downloads.html> in the address bar. The page content includes:

- Full list of changes: [README.txt \(PGP Signature\)](#)
- Checksums:
 - SHA256 sums for VeraCrypt 1.26.7 release files ([PGP Signature](#))
 - SHA512 sums for VeraCrypt 1.26.7 release files ([PGP Signature](#))
- Bundle containing all release files:
 - [VeraCrypt 1.26.7 Bundle.7z \(PGP Signature\)](#)
 - [SHA256/SHA512 sums of VeraCrypt 1.26.7 Bundle \(PGP Signature\)](#)

Previous Versions Archives

The packages and installers of all previous VeraCrypt versions can be found at <https://sourceforge.net/projects/veracrypt/files/>.

Important Notice for VeraCrypt RPM Packages

VeraCrypt signs its RPM packages using the standard RPM signature mechanism which embeds the signature into the packages. This makes verification easier for users. Follow these steps to verify the integrity of the RPM packages:

1. **Import VeraCrypt GPG Public Key:** In a terminal window, use the following command to import the VeraCrypt GPG public key:

```
sudo rpm --import https://www.idrix.fr/VeraCrypt/VeraCrypt_PGP_public_key.asc
```

This process will display the key's fingerprint and ask for your confirmation before adding the key.
2. **Validate the Key's Fingerprint:** During the key import process, carefully verify that the displayed fingerprint matches the official VeraCrypt fingerprint:

5069 A233 D55A 0EEB 174A 5FC3 821A CD02 680D 16DE

This step is vital for your security. Do not proceed if the fingerprints do not match.

By following these steps, you help ensure the integrity of the software you install, safeguarding your system from potential threats.

Remember, your security is paramount. Always verify the authenticity of your downloads.

[What are PGP signatures?](#)

PGP Public Key: https://www.idrix.fr/VeraCrypt/VeraCrypt_PGP_public_key.asc (ID=0x680D16DE, Fingerprint=5069A233D55A0EEB174A5FC3821ACD02680D16DE)

Verifying Files from the Internet (Continue)

- Check sha256sum hash
- Compare with hash on web site
- To be sure that hash on web site wasn't spoofed, check signature
- You can --import Key and **trust** this Key to avoid WARNING

```
▶ 21:27 alexc@Alexs-MBP /Users/alexc
% head -1 veracrypt-1.26.7-sha256sum.txt
5e8ce0dd03aa1a9728157b224a36b622007f27d4ebb04395fb73f23a19ac6114 README.TXT
▶ 21:27 alexc@Alexs-MBP /Users/alexc
% shasum -a 256 README.TXT
5e8ce0dd03aa1a9728157b224a36b622007f27d4ebb04395fb73f23a19ac6114 README.TXT
▶ 21:28 alexc@Alexs-MBP /Users/alexc
% gpg --verify veracrypt-1.26.7-sha256sum.txt.sig
gpg: assuming signed data in 'veracrypt-1.26.7-sha256sum.txt'
gpg: Signature made Mon May 20 13:13:41 2024 PDT
gpg:           using RSA key 5069A233D55A0EEB174A5FC3821ACD02680D16DE
gpg: key 821ACD02680D16DE: public key "VeraCrypt Team (2018 - Supersedes Key ID=0x54DDD393) <veracrypt@idrix.fr>" imported
gpg: Total number processed: 1
gpg:           imported: 1
gpg: Good signature from "VeraCrypt Team (2018 - Supersedes Key ID=0x54DDD393) <veracrypt@idrix.fr>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:           There is no indication that the signature belongs to the owner.
Primary key fingerprint: 5069 A233 D55A 0EEB 174A 5FC3 821A CD02 680D 16DE
```



Encrypt / Decrypt Files

```
gpg -e -s -a -u achesnakov@mailfence.com -r achesnakov@mailfence.com message.txt
```

```
gpg -d message.txt.asc
```

-e, --encrypt

-s, --sign

-a, --armor

-d, --decrypt

-u, --local-user USER-ID

use USER-ID to sign, encr, decr

-r, --recipient USER-ID

encrypt for USER-ID

In this case I encrypted for myself

You can encrypt for different recipient

```
% gpg -e -s -a -u achesnakov@mailfence.com -r achesnakov@mailfence.com message.txt
>22:30 alexc@Alexs-MBP /Users/alexc
% cat message.txt
This is my message.
>22:30 alexc@Alexs-MBP /Users/alexc
% gpg --decrypt message.txt.asc
gpg: encrypted with 256-bit ECDH key, ID 1492A775A41C110D, created 2024-08-16
    "Alex Chesnakov <achesnakov@mailfence.com>"
This is my message.
gpg: Signature made Tue Aug 20 22:30:16 2024 PDT
gpg:                               using EDDSA key 4AA6B476861C007DDE95FC2B1FA31C2070C6150F
gpg:                               issuer "achesnakov@mailfence.com"
gpg: Good signature from "Alex Chesnakov <achesnakov@mailfence.com>" [ultimate]
>22:30 alexc@Alexs-MBP /Users/alexc
% cat message.txt.asc
-----BEGIN PGP MESSAGE-----

hF4DFJKndaQcEQ0SAQdAaRSq0EfYSeh0+jqcTh0C31ntDgLl1FFZtZHFPyLGhEw
14ruMI/KwC9orEv7aHrWmDnewbLayMFQu4juicNQRShnNKTpEhzxUvUJoCc7qgJ
0sAoAU0xsWSWESj15bi/Pu0/iWx/aWZmTQvtFWjV10jA0q7yP3hGECK1STHmBpFK
w1dmBA1MLVVFB1L5kWvQkwsWNZTvpvyUbvLkLchERYZfSI7f8R1m+PX8ChSxErt5z
X0bKHnUMja+09vtctXwyS83kj2ML/eYV3hZ7FazFHUozYiEZzfxOKIoH42o8w7h
MA0LV39ce72AzfwIzsTaGLqateIy4BPsiAvEM11j6qppsofc+Q5ioI9ERaxrzctCLS4Ey8Jj03I3hgBGfMKAQc6okyAfAvsWCT4c4356g5Ly0vvNynTDqcpge
=qD4q
-----END PGP MESSAGE-----
```

Not Covered

- Web of Trust
- Traditional OpenPGP keyservers
- Key Signing Parties
- Photo IDs in OpenPGP keys
- Sub-Keys
- Edit Keys (`gpg --edit-key`)
- Physical Keys: YubiKeys, Smartcards

Q&A

- SSL/TLS and SSH use public / private keys as well. What the difference?

Resources:

- <https://emailselfdefense.fsf.org/en/>
- <https://www.gnupg.org/gph/en/manual.html>
- <https://www.oreilly.com/library/view/pgp-gpg/9781457185212/>
Free O'REILLY access with San Jose Public Library
<https://www.sjpl.org/resource/oreilly/>

Thanks



<https://github.com/olexandrch>



<https://www.linkedin.com/in/alex-chesnakov/>

