


Creating GPG Keys

 In other languages: **English** (en) | **Español** (es) (https://fedoraproject.org/wiki/Creating_GPG_Keys/es)
| **Português do Brasil** (pt-br) (https://fedoraproject.org/wiki/Creating_GPG_Keys/pt-br)
| **Русский** (ru) (https://fedoraproject.org/wiki/Creating_GPG_Keys/ru) |
中文(中国大陆) (zh-cn) (https://fedoraproject.org/wiki/Creating_GPG_Keys/zh-cn)

[edit (https://fedoraproject.org/w/index.php?title=Template:Lang/Creating_GPG_Keys&action=edit)]

This page explains in detail how to obtain a GPG key using common Fedora utilities. It also provides information on managing your key as a Fedora contributor.

Creating GPG Keys

Creating GPG Keys Using the GNOME Desktop

Install the **Seahorse** utility, which makes GPG key management easier. From the main menu, select *Applications > Add/Remove Software*. Select the *Search* tab and enter the name *seahorse*. Select the checkbox next to the *seahorse* package and select *Apply* to add the software. You can also install **Seahorse** using the command line with the command `su -c "yum install seahorse"`.

To create a key, go to the Activities overview and select *Passwords and Encryption Keys*, which starts the application **Seahorse**.

From the *File* menu select *New...* then *PGP Key* then click *Continue*. Type your full name, email address, and an optional comment describing who you are (e.g.: John C. Smith, jsmith@example.com, The Man). Click *Create*. A dialog is displayed asking for a passphrase for the key. Choose a passphrase that is strong but also easy to remember. Click *OK* and the key is created.



If you forget your passphrase, the key cannot be used and any data encrypted using that key will be lost.

To find your GPG key ID click on the *My Personal Keys* tab and look in the *Key ID* column next to the newly created key. In most cases, if you are asked for the key ID, you should prepend "0x" to the key ID, as in "0x6789ABCD".

Now you should make a backup of your private key.

Creating GPG Keys Using the KDE Desktop

Start the **KGpg** program from the main menu by selecting *Utilities > PIM > KGpg*. If you have never used **KGpg** before, the program walks you through the process of creating your own GPG key pair.

A dialog box appears prompting you to create a new key pair. Enter your name, email address, and an optional comment. You can also choose an expiration time for your key, as well as the key strength (number of bits) and algorithms. The next dialog box prompts you for your passphrase. At this point, your key appears in the main **KGpg** window.



If you forget your passphrase, the key cannot be used and any data encrypted using that key will be lost.

To find your GPG key ID, look in the *Key ID* column next to the newly created key. In most cases, if you are asked for the key ID, you should prepend "0x" to the key ID, as in "0x6789ABCD".

Now you should make a backup of your private key.

Creating GPG Keys Using the Command Line

Use the following shell command:

```
gpg --gen-key
```

This command generates a key pair that consists of a public and a private key. Other people use your public key to authenticate and/or decrypt your communications. Distribute your **public** key as widely as possible, especially to people who you know will want to receive authentic communications from you, such as a mailing list. The Fedora

Documentation Project, for example, asks participants to include a GPG public key in their self-introduction .

A series of prompts directs you through the process. Press the **Enter** key to assign a default value if desired. The first prompt asks you to select what kind of key you prefer:

```
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)
Your selection?
```

In almost all cases, the default is the correct choice. A RSA/RSA key allows you not only to sign communications, but also to encrypt files.

Next, choose the key size:

```
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
```

Again, the default is sufficient for almost all users, and represents an *extremely* strong level of security.

Next, choose when the key will expire. It is a good idea to choose an expiration date instead of using the default, which is *none*. If, for example, the email address on the key becomes invalid, an expiration date will remind others to stop using that public key.

```
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)
```

Entering a value of `1y`, for example, makes the key valid for one year. (You may change this expiration date after the key is generated, if you change your mind.)

Before the `gpg` program asks for signature information, the following prompt appears:

```
Is this correct (y/n)?
```

Enter `y` to finish the process.

Next, enter your name and email address. *Remember this process is about authenticating you as a real individual.* For this reason, include your *real name*. Do not use aliases or handles, since these disguise or obfuscate your identity.

Enter your real email address for your GPG key. If you choose a bogus email address, it will be more difficult for others to find your public key. This makes authenticating your communications difficult. If you are using this GPG key for self-introduction on a mailing list, for example, enter the email address you use on that list.

Use the comment field to include aliases or other information. (Some people use different keys for different purposes and identify each key with a comment, such as "Office" or "Open Source Projects.")

At the confirmation prompt, enter the letter **O** to continue if all entries are correct, or use the other options to fix any problems.

Finally, enter a passphrase for your secret key. The `gpg` program asks you to enter your passphrase twice to ensure you made no typing errors.

Finally, `gpg` generates random data to make your key as unique as possible. Move your mouse, type random keys, or perform other tasks on the system during this step to speed up the process. Once this step is finished, your keys are complete and ready to use:

```
pub 1024D/1B2AFA1C 2005-03-31 John Q. Doe (Fedora Docs Project) <jqdoe@example.com>
Key fingerprint = 117C FE83 22EA B843 3E86 6486 4320 545E 1B2A FA1C
sub 1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

The key fingerprint is a shorthand "signature" for your key. It allows you to confirm to others that they have received your actual public key without any tampering. You do not need to write this fingerprint down. To display the fingerprint at any time, use this command, substituting your email address:

```
gpg --fingerprint jqdoe@example.com
```

Your "GPG key ID" consists of 8 hex digits identifying the public key. In the example above, the GPG key ID is 1B2AFA1C. In most cases, if you are asked for the key ID, you should prepend "0x" to the key ID, as in "0x1B2AFA1C".



If you forget your passphrase, the key cannot be used and any data encrypted using that key will be lost.

Now you should make a backup of your private key.

Making a Backup



Make a backup of your private key and put it in a safe location such as a CD, DVD, or USB key stored in a secure container.

Making a Key Backup Using the GNOME Desktop

Right-click your key and select *Properties*. Select the *Details* tab, and *Export*, next to the *Export Complete Key* label. Select a destination filename and click *Save*.

Store the copy in a secure place, such as a locked container. Now you are ready to make your public key available to others.

Making a Key Backup Using the KDE Desktop

Right-click your key and select *Export Secret Key*. At the confirmation dialog, click *Export* to continue, then select a destination filename and click *Save*.

Store the copy in a secure place, such as a locked container. Now you are ready to make your public key available to others.

Making a Key Backup Using the Command Line

Use the following command to make the backup, which you can then copy to a destination of your choice:

```
gpg --export-secret-keys --armor jqdoe@example.com > jqdoe-privkey.asc
```

Store the copy in a secure place, such as a locked container. Now you are ready to make your public key available to others.

Making Your Public Key Available

When you make your public key available to others, they can verify communications you sign, or send you encrypted communications if necessary. This procedure is also known as *exporting*.

You should now export your key using GNOME, KDE, or the command line. You can also copy your key manually to a file if you wish to email it to individuals or groups.

Exporting a GPG Key Using the GNOME Desktop

Export the key to a public keyserver where other project members can obtain it. Right-click the key and select *Sync and Publish Keys...* (or in the seahorse menu bar click on the *Remote* menu and select *Sync and Publish Keys...*). Click *Key Servers*, select *hkp://subkeys.pgp.net:11371* in the *Publish Keys To* combobox, click *Close* and then *Sync*.



If the key can't be synchronized, the traffic may be blocked by a firewall. In this case you can publish the key with a different method. Copy the key by right-clicking the key and selecting *Copy Public Key*. Use a web browser to load the page "<http://subkeys.pgp.net/>", paste (Ctrl+V) the public key in the *Submit a Key* textbox and click *Submit this key to the keyserver!*. Your public key then becomes available on the public server for others to copy and use.

You can now read more about safeguarding your key or use your browser to go back to a previous page.

Exporting a GPG Key Using the KDE Desktop

After your key has been generated, you can export the key to a public keyserver by right-clicking on the key in the main window, and selecting *Export Public Keys*. From there you can export your public key to the clipboard, an ASCII file, to an email, or directly to a key server. Export your public key to the default key server.



If the key can't be synchronized, the traffic may be blocked by a firewall. In this case you can publish the key with a different method. Copy the key by right-clicking the key and selecting *Copy Public Key*. Use a web browser to load the page "http://subkeys.pgp.net/", paste (Ctrl+V) the public key in the *Submit a Key* textbox and click *Submit this key to the keyserver!*. Your public key then becomes available on the public server for others to copy and use.

You can now read more about safeguarding your key or use your browser to go back to a previous page.

Exporting a GPG Key Using the Command Line

Use the following command to send your key to a public keyserver:

```
gpg --send-key KEYNAME
```

For *KEYNAME*, substitute the key ID or fingerprint of your primary keypair.

This will send your key to the gnupg default key server (keys.gnupg.net), if you prefer another one use :

```
gpg --keyserver hkp://pgp.mit.edu --send-key KEYNAME
```

Replacing "pgp.mit.edu" with your server of choice.

You can now read more about safeguarding your key or use your browser to go back to a previous page.

Copying a Public Key Manually

If you want to give or send a file copy of your key to someone, use this command to write it to an ASCII text file:

```
gpg --export --armor jqdoe@example.com > jqdoe-pubkey.asc
```

You can now read more about safeguarding your key or use your browser to go back to a previous page.

Safeguarding Your Secret Key

Treat your secret key as you would any very important document or physical key. (Some people always keep their secret key on their person, either on magnetic or flash media.) If you lose your secret key, you will be unable to sign communications, or to open encrypted communications that were sent to you.



Always keep your passphrase secret!

Even if your secret key is accessed by someone else, they will be unable to use it without your passphrase. Do not choose a passphrase that someone else might easily guess. Do not use single words (in any language), strings of numbers such as your telephone number or an official document number, or biographical data about yourself or your family for a passphrase. The most secure passphrases are *very long* and contain a mixture of uppercase and lowercase letters, numbers, digits, and symbols. Choose a passphrase that you will be able to remember, however, since writing this passphrase down anywhere makes it immediately less secure.

GPG Key Revocation

When you revoke a key, you withdraw it from public use. *You should only have to do this if it is compromised or lost, or you forget the passphrase.*

Generating a Revocation Certificate

When you create the key pair you should also create a key revocation certificate. If you later issue the revocation certificate, it notifies others that the public key is not to be used. Users may still use a revoked public key to verify old signatures, but not encrypt messages. As long as you still have access to the private key, messages received previously may still be decrypted. If you forget the passphrase, you will not be able to decrypt messages encrypted to that key.

```
gpg --output revoke.asc --gen-revoke KEYNAME
```

If you do not use the `--output` flag, the certificate will print to standard output.

For *KEYNAME*, substitute either the key ID of your primary keypair or any part of a user ID that identifies your keypair. Once you create the certificate (the `revoke.asc` file), you should protect it. If it is published by accident or through the malicious actions of others, the public key will become unusable. It is a good idea to write the revocation certificate to secure removable media or print out a hard copy for secure storage to maintain secrecy.

Revoking a key

```
gpg --import revoke.asc
```

Once you locally revoke the key, you should send the revoked certificate to a key server, regardless of whether the key was originally issued in this way. Distribution through a server helps other users to quickly become aware the key has been compromised.

Export to a key server with the following command:

```
gpg --keyserver subkeys.pgp.net --send KEYNAME
```

For *KEYNAME*, substitute either the key ID of your primary keypair or any part of a user ID that identifies your keypair.

See the [Using_GPG](#) page for more ideas on using your new GPG keys.

Retrieved from "https://fedoraproject.org/w/index.php?title=Creating_GPG_Keys&oldid=377257"

Categories: Informal Documentation | Encryption

Copyright © 2016 Red Hat, Inc. and others. All Rights Reserved. For comments or queries, please contact us.
The Fedora Project is maintained and driven by the community and sponsored by Red Hat. This is a community maintained site. Red Hat is not responsible for content.

This page was last modified on 30 April 2014, at 11:28. | Content is available under Attribution-Share Alike 3.0 Unported unless otherwise noted.

| [Sponsors](#) | [Legal](#) | [Trademark Guidelines](#)