

**Improve the Security of Your OpenPGP Key by Using Subkeys**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Revised by** | **Comments** |
| 0.11 | 20131230 | RCS | Change references to "GPG subkeys" to "OpenPGP subkeys" (Thanks to Daniel Kahn Gillmor for the pointer) |
| 0.1 | 20131217 | RCS | Updated and improved content; added CC license |
| 0.0 | 20100519 | RCS | Initial draft |

**Copyright**

This article is copyright © 2010,2013 [Connexer Ltd.](http://www.connexer.com/)

**Licensing**

[Creative Commons License](http://creativecommons.org/licenses/by-sa/4.0/)  
This article is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-sa/4.0/).

**Table of Contents**

[Introduction](http://www.connexer.com/articles/openpgp-subkeys#intro)

[What Are Subkeys?](http://www.connexer.com/articles/openpgp-subkeys#subkeys)

[Advantages](http://www.connexer.com/articles/openpgp-subkeys#advantages)

[Disadvantages](http://www.connexer.com/articles/openpgp-subkeys#disadvantages)

[Tweaking This Guide](http://www.connexer.com/articles/openpgp-subkeys#tweaking)

[Caveat](http://www.connexer.com/articles/openpgp-subkeys#caveat)

[Step-by-step Guide](http://www.connexer.com/articles/openpgp-subkeys#stepbystep)

**Introduction**

The information in this article was adapted from "Using multiple subkeys in GPG" by Adrian von Bidder <avbidder+gpg@fortytwo.ch> (original URL: <http://fortytwo.ch/gpg/subkeys>).

The purpose of this article is to make it easier for the reader to more securely manage his or her GPG key through the use of OpenPGP subkeys. Understanding that this can potentially make overall management of your GPG key more challenging, there are very good reasons to take an approach like this. There are also some drawbacks. This article will briefly discuss the advantages and disadvantages.

**What Are Subkeys?**

First, it is worth explaining what subkeys are. Subkeys are a feature of OpenPGP which allows keys which are subordinate to the main OpenPGP key. They can be identified with their own key fingerprint, but are most often associated with the fingerprint of the main key. Please note that this is not meant to be a comprehensive or exhaustive explanation of the concept of subkeys, but rather touches only on the points important to the author. There is an abundance of documentation on all aspects of OpenPGP and GPG, which the reader is encouraged to consult for a more thorough understanding. There are two main uses of OpenPGP: encryption, and signing. When you create an OpenPGP key with GPG (which is the focus of this article), you create two subkeys by default: one for encryption and one for signing. The subkeys discussed in this article are only used for signing. The reason for this is that it is impossible to sensibly employ multiple encryption subkeys. A simple explanation for this is that when someone goes to send you an encrypted message, that person has no way of knowing which possible subkey (assuming you have more than one) ought to be used to encrypt the message. The only sensible thing would be to distribute the public part of every encryption subkey and then anybody encrypting anything to you would need to send you the same message once for each encryption subkey you have created. This could certainly be problematic. Having said that, implementing the approach detailed in this article will not provide any additional protection to your encryption subkey. That means that if your device is lost or compromised, you would have a concern that whomever acquired your GPG key could potentially decrypt anything encrypted to you. There are ways around this (e.g., using a smart card or other hardware token) but they are beyond the scope of this article. As far as signing subkeys, it is very simple (from an infrastructure perspective) to handle multiple subkeys. They all associate to their primary key, so anything signed by any subkey can be associated with the main key and can hence be considered to be signed by the same person.

**Advantages**

There are a number of advantages to employing the approach documented here to managing your GPG key. The principal reason is that should you have your GPG secret key loaded on a mobile device and that device is stolen, you will find yourself with a serious problem. If you do not have a backup of your key stored somewhere, you are stuck generating a new key. Even if you do have a backup somewhere, you will be left wondering if the key on the mobile device has been compromised in some way. Even in a controlled situation (e.g., sending a laptop in for warranty repair), as opposed to having a device stolen, letting someone else have control of your secret key may be unacceptable. By employing the approach detailed in this article, if a subkey is lost or compromised, it can simply and be revoked and a new one issued without having to generate and entirely new GPG key. This results in significantly less impact to the web of trust. Another advantage to the approach detailed in this article is that storing the primary secret key on offline removable media means that it is impossible for a system or device compromise to result in a compromise of the primary secret key. Of course, the media containing the primary secret key would need to be properly protected.

**Disadvantages**

It is also important to note that there are some disadvantages. Keeping the primary secret key on removable media makes signing other GPG keys problematic, since the primary secret key must be loaded in order to do the signing. Additionally, using expiration dates with the subkeys (as is recommended in this article) means that the subkeys must be extended prior to expiration, or new subkeys issued if they are allowed to expire. The approach that I take is that I make subkeys good for one year, and queue up all the keys which I will sign until 60-90 days before the subkeys will expire. At that point, I will load the primary secret key, sign all the keys I have queued up throughout the year and then extend the expiration dates of the subkeys at the same time. Note that I do not place an expiration date on the primary key. I want it to be good forever. Occasionally someone will request that I sign their key right away and then I will have to go through part of this exercise. This article also recommends using different passphrases for each subkey, which could be a headache if you have many passwords and passphrases to remember. Another disadvantage is that other people who use your public key (i.e., to verify something you have signed) may receive errors about your key being expired if they do not regularly update their GPG keyrings based on the public keyservers.

**Tweaking This Guide**

Please note that the approach that detailed in this article does not employ every possible tactic that a truly paranoid person would likely want to employ. For example, at one point the steps below will have you remove various secret keys and transfer them to your different systems using scp. If you are really paranoid, you will want to employ utilities like wipe(1) and you will probably want to use something other than scp to move the keys. Additionally, this article has you do certain things which you may think are too extreme, like placing your primary secret key onto a removable media and removing it entirely from your systems so that it is only stored offline. You may choose to modify the approaches recommended here to suit your specific needs.

**Caveat**

There is also one important caveat with regards to the output shown in this article. They keys I used at the time I initially wrote this article are considered insecure by more modern standards. When I initially created my GPG key in 2005, it was common to use 1024 bit DSA keys for signing and 2048 bit keys for encryption. Those were the GPG defaults, if I recall correctly. I have since discontinued using these keys and have upgraded to 4096 bit RSA keys for both signing and encryption. That said, the procedure remains the same, even if the output has changed slightly.

**Step-by-step Guide**

What follows below is a step-by-step guide to implementing the subkey management approach summarized above. This article assumes that you have already created the subkeys which you wish to manage and that you already know to which systems you wish to transfer them.

1. Create a backup of your GPG directory:
2. cp -a ~/.gnupg ~/.gnupg.orig

If you are doing this for the first time, i.e., you have just created your subkeys, then you can skip over steps 2 through 5, as those are only relevant if you have already done this once. In which case you have stored your primary secret key to removable media and then need to reload it in order to proceed with the subkey renewal and other steps detailed below.

1. Check for the presence of the secret key:
2. roberto@desktop:~$ gpg --list-secret-keys
3. /network/home/roberto/.gnupg/secring.gpg
4. ----------------------------------------
5. sec# 1024D/B2B97BB1 2005-10-01
6. uid Roberto C. Sanchez <roberto@familiasanchez.net>
7. uid Roberto C. Sanchez <sanchezr@ieee.org>
8. uid Roberto C. Sanchez <roberto@connexer.com>
9. ssb 2048g/D27DA3A2 2005-10-01
10. ssb 1024D/A7C89464 2006-07-15

In my case, I have three subkeys which I manage. One subkey is for use on my home/office network, while each of the other two subkeys are for my two laptops. In the listing above, the primary key is listed first, with each of its associated user IDs (GPG keys can multiple identities associated with them), followed by the encryption subkey, 2048g/D27DA3A2 in this case, and one of my signing subkeys, 1024D/A7C89464. The hash (#) after "sec" listed before the primary key indicates that the secret key is not present. The first key listed is always the primary key. So, in this instance, only the public part of the primary key is present. The dates correspond to the creation date of each key or subkey.

1. Mount the removable media containing your primary secret key:
2. mount /mnt/cdrom/
3. Copy only the secret ring from your removable media into your GPG home directory:
4. cp /mnt/cdrom/gnupg.bak/secring.gpg ~/.gnupg/

The reason to copy in only the secret key is that if you have taken the approach of storing your primary secret key on removable media you may have updates to your public keyring that you do not want to overwrite. For example, if you have retrieved new public keys or if GPG has performed trustdb updates. This ensures that you don't lose those updates. Toward the end of this procedure you will backup your updated GPG home directory with all the secret keys and all the updated expiration dates.

1. Check for the presence of the secret key:
2. roberto@desktop:~$ gpg --list-secret-keys
3. /network/home/roberto/.gnupg/secring.gpg
4. ----------------------------------------
5. sec 1024D/B2B97BB1 2005-10-01
6. uid Roberto C. Sanchez <roberto@familiasanchez.net>
7. uid Roberto C. Sanchez <sanchezr@ieee.org>
8. uid Roberto C. Sanchez <roberto@connexer.com>
9. ssb 2048g/D27DA3A2 2005-10-01
10. ssb 1024D/42B5A9F6 2006-07-15
11. ssb 1024D/A7C89464 2006-07-15
12. ssb 1024D/C0050B4F 2006-07-15

There is no hash (#) after the "sec" corresponding to the first key, indicating that both public and secret parts of the key are present. The end of the output indicates that the encryption subkey and all three signing subkeys are present.

1. Set new expiration dates:
2. roberto@desktop:~$ gpg --edit-key B2B97BB1
3. Secret key is available.
5. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
6. trust: ultimate validity: ultimate
7. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
8. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
9. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
10. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
11. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
12. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
13. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
15. Command> key 2
17. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
18. trust: ultimate validity: ultimate
19. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
20. sub\* 1024D/A7C89464 created: 2006-07-15 expires: 2008-07-15 usage: S
21. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2008-07-15 usage: S
22. sub 1024D/C0050B4F created: 2006-07-15 expires: 2008-07-15 usage: S
23. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
24. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
25. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
27. Command> expire
28. Changing expiration time for a subkey.
29. Please specify how long the key should be valid.
30. 0 = key does not expire
31. <n> = key expires in n days
32. <n>w = key expires in n weeks
33. <n>m = key expires in n months
34. <n>y = key expires in n years
35. Is this correct? (y/N)
36. Key is valid for? (0) 384
37. Key expires at Wed 15 Jul 2009 07:46:39 PM EDT
38. Is this correct? (y/N) y
40. You need a passphrase to unlock the secret key for
41. user: "Roberto C. Sanchez <roberto@connexer.com>"
42. 1024-bit DSA key, ID B2B97BB1, created 2005-10-01

45. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
46. trust: ultimate validity: ultimate
47. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
48. sub\* 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
49. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2008-07-15 usage: S
50. sub 1024D/C0050B4F created: 2006-07-15 expires: 2008-07-15 usage: S
51. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
52. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
53. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
55. Command> key 2
57. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
58. trust: ultimate validity: ultimate
59. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
60. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
61. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2008-07-15 usage: S
62. sub 1024D/C0050B4F created: 2006-07-15 expires: 2008-07-15 usage: S
63. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
64. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
65. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
67. Command> key 3
69. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
70. trust: ultimate validity: ultimate
71. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
72. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
73. sub\* 1024D/42B5A9F6 created: 2006-07-15 expires: 2008-07-15 usage: S
74. sub 1024D/C0050B4F created: 2006-07-15 expires: 2008-07-15 usage: S
75. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
76. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
77. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
79. Command> expire
80. Changing expiration time for a subkey.
81. Please specify how long the key should be valid.
82. 0 = key does not expire
83. <n> = key expires in n days
84. <n>w = key expires in n weeks
85. <n>m = key expires in n months
86. <n>y = key expires in n years
87. Key is valid for? (0) 384
88. Key expires at Wed 15 Jul 2009 07:47:07 PM EDT
89. Is this correct? (y/N) y
91. You need a passphrase to unlock the secret key for
92. user: "Roberto C. Sanchez <roberto@connexer.com>"
93. 1024-bit DSA key, ID B2B97BB1, created 2005-10-01

96. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
97. trust: ultimate validity: ultimate
98. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
99. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
100. sub\* 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
101. sub 1024D/C0050B4F created: 2006-07-15 expires: 2008-07-15 usage: S
102. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
103. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
104. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
106. Command> key 3
108. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
109. trust: ultimate validity: ultimate
110. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
111. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
112. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
113. sub 1024D/C0050B4F created: 2006-07-15 expires: 2008-07-15 usage: S
114. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
115. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
116. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
118. Command> key 4
120. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
121. trust: ultimate validity: ultimate
122. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
123. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
124. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
125. sub\* 1024D/C0050B4F created: 2006-07-15 expires: 2008-07-15 usage: S
126. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
127. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
128. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
130. Command> expire
131. Changing expiration time for a subkey.
132. Please specify how long the key should be valid.
133. 0 = key does not expire
134. <n> = key expires in n days
135. <n>w = key expires in n weeks
136. <n>m = key expires in n months
137. <n>y = key expires in n years
138. Key is valid for? (0) 384
139. Key expires at Wed 15 Jul 2009 07:47:32 PM EDT
140. Is this correct? (y/N) y
142. You need a passphrase to unlock the secret key for
143. user: "Roberto C. Sanchez <roberto@connexer.com>"
144. 1024-bit DSA key, ID B2B97BB1, created 2005-10-01

147. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
148. trust: ultimate validity: ultimate
149. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
150. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
151. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
152. sub\* 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
153. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
154. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
155. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
157. Command> key 4
159. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
160. trust: ultimate validity: ultimate
161. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
162. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
163. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
164. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
165. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
166. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
167. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
169. Command> Save changes? (y/N) y

You will note that I did not use 365 for the number of days until the next expiration. That is because the subkeys were set to expire on the 15th of July, and I wanted to extend them for exactly one year. Since I was going through this exercise nearly a month prior to the expiration, I needed to add the number of days remaining until the original expiration to get the extension to be exactly one year.

1. Backup your GPG home directory with all the secret keys in it:
2. cp -a ~/.gnupg ~/.gnupg.bak

This backup will be used several times as it contains the full secret keyring with all the secret keys (for both the primary and subkeys).

1. Export the public key and primary secret key:
2. roberto@desktop:~$ gpg --export B2B97BB1 >pubkey.gpg
3. roberto@desktop:~$ file pubkey.gpg
4. pubkey.gpg: GPG key public ring
5. roberto@desktop:~$ gpg --export-secret-keys B2B97BB1 >secret\_primary\_key.gpg
6. roberto@desktop:~$ file secret\_primary\_key.gpg
7. secret\_primary\_key.gpg: PGP key security ring

The exported public key can be posted on a website or in some other public place to allow others to verify things that you have signed or to encrypt something to you. A later step also uploads the keys to a public key server.

1. Delete the second and third subkeys:
2. roberto@desktop:~$ gpg --edit-key B2B97BB1
3. Secret key is available.
5. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
6. trust: ultimate validity: ultimate
7. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
8. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
9. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
10. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
11. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
12. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
13. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
15. Command> key 3
17. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
18. trust: ultimate validity: ultimate
19. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
20. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
21. sub\* 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
22. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
23. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
24. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
25. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
27. Command> key 4
29. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
30. trust: ultimate validity: ultimate
31. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
32. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
33. sub\* 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
34. sub\* 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
35. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
36. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
37. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
39. Command> delkey
40. Do you really want to delete the selected keys? (y/N) y
42. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
43. trust: ultimate validity: ultimate
44. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
45. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
46. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
47. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
48. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
50. Command> Save changes? (y/N) y

This step makes it possible to export just the secret part of the first subkey. This is important because having multiple subkeys does you no good if you put the secret part of every subkey in every place. The maximum benefit is realized by placing only a single secret subkey on each device. Then a loss or compromise of a device only necessitates revoking and replacing a single subkey.

1. Export the single remaining subkey:
2. roberto@desktop:~$ gpg --export-secret-subkeys B2B97BB1 >desktop\_secret\_subkey.gpg
3. Remove the GPG home directory that only contains a single secret subkey:
4. roberto@desktop:~$ rm -rf ~/.gnupg
5. Restore the backup GPG home directory with all the secret keys:
6. roberto@desktop:~$ cp -a ~/.gnupg.bak ~/.gnupg
7. Delete the first and third subkeys:
8. roberto@desktop:~$ gpg --edit-key B2B97BB1
9. Secret key is available.
11. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
12. trust: ultimate validity: ultimate
13. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
14. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
15. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
16. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
17. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
18. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
19. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
21. Command> key 2
23. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
24. trust: ultimate validity: ultimate
25. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
26. sub\* 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
27. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
28. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
29. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
30. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
31. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
33. Command> key 4
35. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
36. trust: ultimate validity: ultimate
37. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
38. sub\* 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
39. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
40. sub\* 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
41. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
42. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
43. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
45. Command> delkey
46. Do you really want to delete the selected keys? (y/N) y
48. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
49. trust: ultimate validity: ultimate
50. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
51. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
52. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
53. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
54. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
56. Command> Save changes? (y/N) y
57. Export the single remaining subkey:
58. roberto@desktop:~$ gpg --export-secret-subkeys B2B97BB1 >laptop1\_secret\_subkey.gpg
59. Repeat 11 through 14 for other subkeys.
60. Remove the GPG home directory that only contains a single secret subkey:
61. roberto@desktop:~$ rm -rf ~/.gnupg
62. Restore the backup GPG home directory with all the secret keys:
63. roberto@desktop:~$ cp -a ~/.gnupg.bak/ ~/.gnupg
64. Update public key on keyservers:
65. roberto@desktop:~$ gpg --send-keys B2B97BB1
66. gpg: sending key B2B97BB1 to hkp server subkeys.pgp.net
67. Create a backup of all the pertinent parts:
68. roberto@desktop:~$ mkdir burn
69. roberto@desktop:~$ cp -a ~/.gnupg burn/gnupg.bak
70. roberto@desktop:~$ cp \*.gpg burn/
71. roberto@desktop:~$ genisoimage -J -r -V GPG\ backup -o gpg.iso burn
72. I: -input-charset not specified, using utf-8 (detected in locale settings)
73. Unknown file type (unallocated) burn/.. - ignoring and continuing.
74. Using PUBRI000.GPG;1 for burn/gnupg.bak/pubring.gpg~ (pubring.gpg)
75. Using GPG\_A000.;1 for burn/gnupg.bak/gpg-agent-info (gpg-agent-info-desktop)
76. 70.91% done, estimate finish Thu Jun 26 20:22:04 2008
77. Total translation table size: 0
78. Total rockridge attributes bytes: 2005
79. Total directory bytes: 5162
80. Path table size(bytes): 46
81. Max brk space used 0
82. 7057 extents written (13 MB)

You can create the ISO image and burn it to CD/DVD media or you can copy the files to a USB device, or whatever backup medium you prefer.

1. Prepare the new GPG home directories:
2. roberto@desktop:~$ cp -a ~/.gnupg ~/.gnupg.laptop1
3. roberto@desktop:~$ cp -a ~/.gnupg ~/.gnupg.laptop2
4. roberto@desktop:~$ cp desktop\_secret\_subkey.gpg ~/.gnupg/secring.gpg
5. cp: overwrite `.gnupg/secring.gpg'? y
6. roberto@desktop:~$ cp laptop2\_secret\_subkey.gpg ~/.gnupg.laptop2/secring.gpg
7. cp: overwrite `.gnupg.laptop2/secring.gpg'? y
8. roberto@desktop:~$ cp laptop1\_secret\_subkey.gpg ~/.gnupg.laptop1/secring.gpg
9. cp: overwrite `.gnupg.laptop1/secring.gpg'? y

This step creates the GPG home directories for the two laptops and copies into each GPG home directory (including the one for the current host) the secret keyrings that contain only a single subkey secret key and omit the primary secret key. The fully populated secret key is preserved in the burn/ directory.

1. Verify the primary secret key is gone and only the secret subkey remains:
2. roberto@desktop:~$ gpg --list-secret-keys
3. /network/home/roberto/.gnupg/secring.gpg
4. ----------------------------------------
5. sec# 1024D/B2B97BB1 2005-10-01
6. uid Roberto C. Sanchez <roberto@familiasanchez.net>
7. uid Roberto C. Sanchez <sanchezr@ieee.org>
8. uid Roberto C. Sanchez <roberto@connexer.com>
9. ssb 2048g/D27DA3A2 2005-10-01
10. ssb 1024D/A7C89464 2006-07-15

Recall, you are looking for the hash (#) after the "sec" associated with the first subkey to indicate that the secret part of the primary key is not present.

1. Set the new passphrases:
2. roberto@desktop:~$ gpg --homedir ~/.gnupg.laptop1 --edit-key B2B97BB1 passwd
3. Secret key is available.
5. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
6. trust: ultimate validity: ultimate
7. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
8. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
9. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
10. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
11. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
12. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
13. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
15. Secret parts of primary key are not available.
17. You need a passphrase to unlock the secret key for
18. user: "Roberto C. Sanchez <roberto@connexer.com>"
19. 2048-bit ELG-E key, ID D27DA3A2, created 2005-10-01
21. Enter the new passphrase for this secret key.

24. Command> Save changes? (y/N) y
25. roberto@desktop:~$ gpg --homedir ~/.gnupg.laptop2 --edit-key B2B97BB1 passwd
26. Secret key is available.
28. pub 1024D/B2B97BB1 created: 2005-10-01 expires: never usage: SC
29. trust: ultimate validity: ultimate
30. sub 2048g/D27DA3A2 created: 2005-10-01 expires: never usage: E
31. sub 1024D/A7C89464 created: 2006-07-15 expires: 2009-07-15 usage: S
32. sub 1024D/42B5A9F6 created: 2006-07-15 expires: 2009-07-15 usage: S
33. sub 1024D/C0050B4F created: 2006-07-15 expires: 2009-07-15 usage: S
34. [ultimate] (1). Roberto C. Sanchez <roberto@connexer.com>
35. [ultimate] (2) Roberto C. Sanchez <roberto@familiasanchez.net>
36. [ultimate] (3) Roberto C. Sanchez <sanchezr@ieee.org>
38. Secret parts of primary key are not available.
40. You need a passphrase to unlock the secret key for
41. user: "Roberto C. Sanchez <roberto@connexer.com>"
42. 2048-bit ELG-E key, ID D27DA3A2, created 2005-10-01
44. Enter the new passphrase for this secret key.

47. Command> Save changes? (y/N) y

I consider this step to be very important. While simply using subkeys helps protect you against someone impersonating you if your device is lost or compromised, it is theoretically possible for that person to brute force your passphrase. For that reason, I use a different passphrase for each keyring after I have separated out the secret subkeys. That way, each of the three (in my case) secret keyrings has only my encryption subkey and one signing subkey, along with having a different passphrase than my other secret keyrings. So, in the event of a compromise, I revoke the compromised subkey and issue a new one with a new passphrase.

1. Transfer each new GPG home directory over to its respective host:
2. roberto@desktop:~$ scp -r ~/.gnupg.laptop1/\* laptop1:~/.gnupg/
3. gpg-agent-info 100% 75 0.1KB/s 00:00
4. gpg-agent-info-desktop 100% 75 0.1KB/s 00:00
5. gpg-agent.conf 100% 23 0.0KB/s 00:00
6. gpg.conf 100% 8096 7.9KB/s 00:00
7. pub\_key.asc 100% 6004 5.9KB/s 00:00
8. pubring.gpg 100% 6829KB 6.7MB/s 00:01
9. pubring.gpg~ 100% 6829KB 3.3MB/s 00:02
10. random\_seed 100% 600 0.6KB/s 00:00
11. revcert-B2B97BB1.asc 100% 264 0.3KB/s 00:00
12. secring.gpg 100% 2252 2.2KB/s 00:00
13. trustdb.gpg 100% 30KB 30.4KB/s 00:00
14. roberto@desktop:~$ scp -r ~/.gnupg.laptop2/\* laptop2:~/.gnupg/
15. gpg-agent-info 100% 75 0.1KB/s 00:00
16. gpg-agent-info-desktop 100% 75 0.1KB/s 00:00
17. gpg-agent.conf 100% 23 0.0KB/s 00:00
18. gpg.conf 100% 8096 7.9KB/s 00:00
19. pub\_key.asc 100% 6004 5.9KB/s 00:00
20. pubring.gpg 100% 6829KB 6.7MB/s 00:00
21. pubring.gpg~ 100% 6829KB 6.7MB/s 00:01
22. random\_seed 100% 600 0.6KB/s 00:00
23. revcert-B2B97BB1.asc 100% 264 0.3KB/s 00:00
24. secring.gpg 100% 2252 2.2KB/s 00:00
25. trustdb.gpg 100% 30KB 30.4KB/s 00:00

If you are paranoid, you will likely not want to use scp to transfer the GPG home directories.

If you managed to read this whole thing, please [let me know](mailto:roberto@connexer.com) what you thought, whether it was useful, or if you have any suggestions for improvement.

Copyright © 2005-2017 [Connexer Ltd.](http://www.connexer.com/)

[Valid XHTML 1.0!](http://validator.w3.org/check?uri=referer) [Valid CSS!](http://jigsaw.w3.org/css-validator/check/referer)

[**Articles**](http://www.connexer.com/articles)

[- Name and Email Validation: You Are Doing It Wrong](http://www.connexer.com/articles/name-and-email-validation)[- How to Ship Your Own Self-Signed Certificate Authority](http://www.connexer.com/articles/ship-your-self-signed-ca)[- Improve the Security of Your OpenPGP Key by Using Subkeys](http://www.connexer.com/articles/openpgp-subkeys)[- Redeliver the Contents of an mbox](http://www.connexer.com/articles/mbox-redeliver)[- Customizing Debian Packages](http://www.connexer.com/articles/debcustomize)[- Setting up a Debian Package Repository](http://www.connexer.com/articles/debrepository)

[More useful content ...](http://www.connexer.com/articles)

[Top of Page](http://www.connexer.com/articles/openpgp-subkeys)

[**Home**](http://www.connexer.com/)

[**Consulting Services**](http://www.connexer.com/consulting)[**Software Development**](http://www.connexer.com/development)[**Open Source Customization**](http://www.connexer.com/customization)

[**About Connexer Ltd.**](http://www.connexer.com/about)[**Contact Information**](http://www.connexer.com/contact)[**Privacy Policy**](http://www.connexer.com/privacy)