

# The KGpg Handbook

Jean-Baptiste Mardelle  
Rolf Eike Beer



# The KGpg Handbook

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Getting Started</b>	<b>6</b>
<b>3</b>	<b>Using KGpg</b>	<b>8</b>
3.1	Generating a key . . . . .	8
3.2	Revoking a key . . . . .	9
3.3	Encrypting Your Data . . . . .	9
3.3.1	Encrypting a file from Konqueror or Dolphin . . . . .	9
3.3.2	Encrypting a text with KGpg's applet . . . . .	10
3.3.3	Encrypting text from KGpg's editor . . . . .	10
3.4	Decrypting Your Data . . . . .	11
3.4.1	Decrypting a file from Konqueror or Dolphin . . . . .	11
3.4.2	Decrypting text with KGpg's applet . . . . .	11
3.4.3	Decrypting a text from the editor . . . . .	11
3.5	Key Management . . . . .	11
3.5.1	Key Manager . . . . .	11
3.5.2	Key properties . . . . .	12
3.5.3	Signing keys . . . . .	12
3.6	Working with key servers . . . . .	15
3.6.1	Communication with key servers . . . . .	15
3.6.2	Key server search results . . . . .	16
3.7	Configuring KGpg . . . . .	16
3.7.1	Encryption . . . . .	17
3.7.2	Decryption . . . . .	17
3.7.3	Appearance . . . . .	18
3.7.4	GnuPG Settings . . . . .	18
3.7.5	Key Servers . . . . .	18
3.7.6	Misc . . . . .	18
<b>4</b>	<b>Credits and License</b>	<b>19</b>

### **Abstract**

KGpg is a simple graphical interface for GnuPG (<http://gnupg.org>).

# Chapter 1

## Introduction

KGpg is a simple interface for GnuPG, a powerful encryption utility. GnuPG (also known as gpg) is included in most distributions and should be installed on your system. You can get the latest version on <http://gnupg.org>.

With KGpg you will be able to encrypt and decrypt your files and emails, allowing much more secure communications. A mini howto on encryption with gpg is available on [GnuPG's web site](#).

With KGpg, you don't need to remember gpg's command lines and options. Almost everything can be done with a few mouse clicks.

## Chapter 2

# Getting Started

Here is a list of KGpg’s main components:

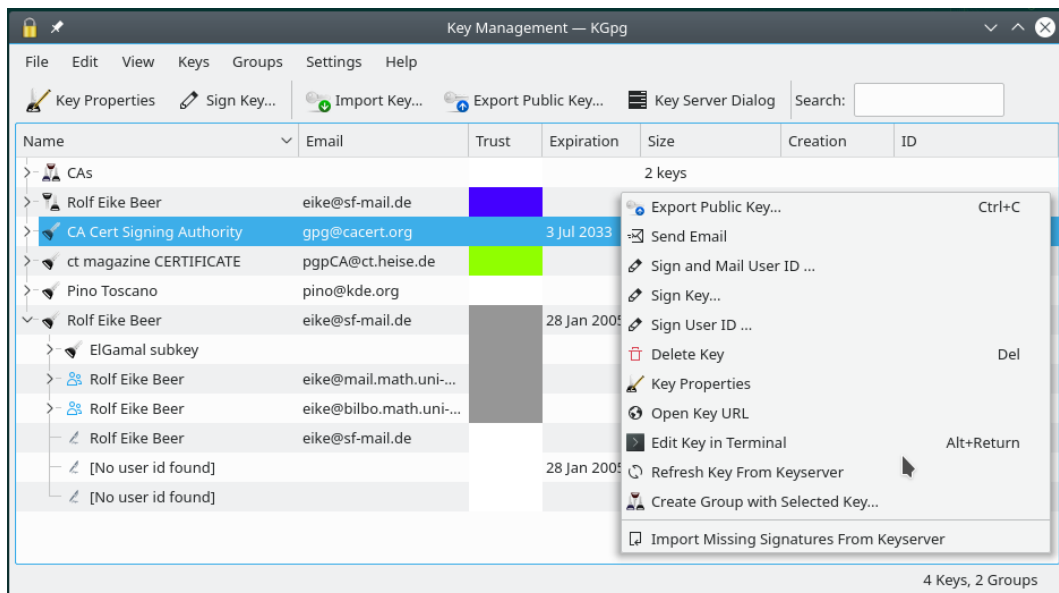
### System Tray Icon



When you start KGpg, a system tray icon will appear. A left mouse button click will open the Key Manager window, while a right mouse button click will open a menu allowing quick access to some important features. If you prefer other options you can change the left mouse button action to show the editor or completely disable the system tray icon using the [settings dialog](#).

Please note that the system tray icon of KGpg is marked as “inactive” basically all the time. Since the system tray applet will usually hide inactive icons the one of KGpg will not be shown until you explicitly request it. For details please have a look at the Plasma documentation.

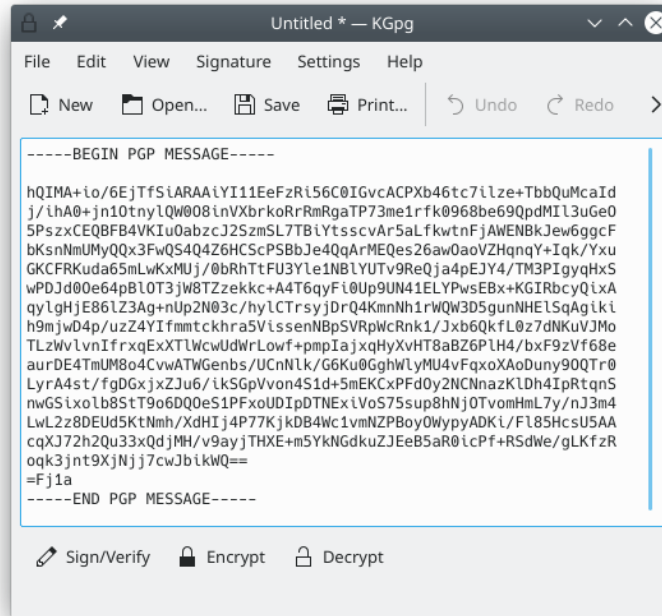
### Key Manager Window



## The KGpg Handbook

That's the central place to manage your keys. To open the [Key Manager window](#), click with the left mouse button on KGpg's applet. You can import, export, sign and edit your keys. Most actions can be performed with a right mouse button click on a key.

### Editor Window



It's a simple text editor, where you can type or paste text to encrypt/decrypt it. To open the [editor](#), click with the right mouse button on KGpg's applet.

### File manager integration

KGpg is integrated in Konqueror and Dolphin. It means that when you right click on a file, you can choose **Actions** → **Encrypt File** to encrypt a file. You can decrypt a file with a left mouse button click.

## Chapter 3

# Using KGpg

There are two ways to encrypt your data:

- Symmetrical encryption: your data is just encrypted with a password. Anybody who has a computer with gpg can decrypt your message if you give him/her the password. To perform a symmetrical encryption, choose “symmetrical encryption” in the options box when asked to choose an encryption key.
- Key encryption: you must first create your key pair (secret key and public key) and give a passphrase. Keep your secret key in a safe place, and exchange your public key with your friends. Then, if you want to send an encrypted message to Alex, you must encrypt the message with Alex’s public key. To decrypt the message, the recipient will need Alex’s secret key and passphrase.

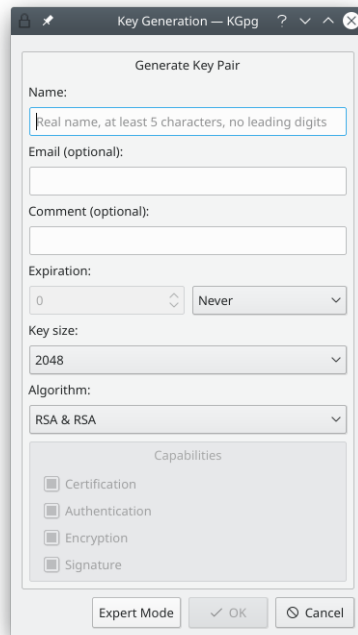
Key encryption is a bit more complicated (you must exchange keys with your friends) but safer. Remember that if you encrypt a message with someone else’s key, you will not be able to decrypt it. You can only decrypt messages that have been encrypted with your public key.

### 3.1 Generating a key

If you don’t have a key, KGpg will automatically pop up the key generation dialog at the first startup. You can also access it in the Key Manager from **Keys** → **Generate Key Pair**.



## The KGpg Handbook



Simply enter your name, Email address and click **Ok**. This will generate a standard gpg key. If you want more options, you can click on the **Expert Mode** button, which will bring up a Konsole window with all of gpg's options.

Many people play around with their first key, generate bad user ids, add comments they later regret or simply forget their passphrase. To avoid such keys to stay valid forever it's usually a good idea to limit the lifetime to some 12 month. You can modify the lifetime of your secret keys later using the [key properties window](#).

### 3.2 Revoking a key

A key pair that has expired can be brought back into an operational state as long as you have access to the private key and the passphrase. To reliably render a key unusable you need to revoke it. Revoking is done by adding a special revocation signature to the key.

This revocation signature can be created together with the key. In this case it is stored in a separate file. This file can later be imported into the keyring and is then attached to the key rendering it unusable. Please note that to import this signature to the key no password is required. Therefore you should store this revocation signature in a safe place, usually one that is different from you key pair. It is a good advise to use a place that is detached from your computer, either copy it to an external storage device like an USB stick or print it out.

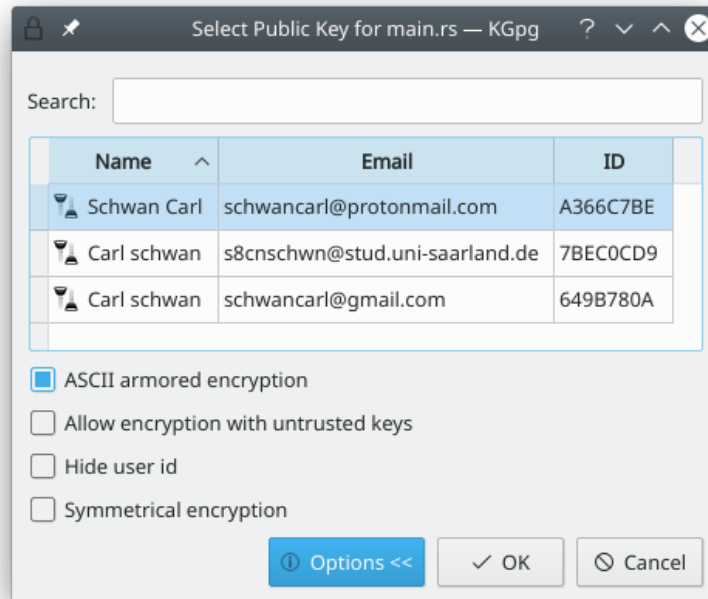
If you have not created such a detached revocation on key creation you can create such a revocation signature at any time choosing **Keys** → **Revoke key**, optionally importing it to your keyring immediately.

### 3.3 Encrypting Your Data

#### 3.3.1 Encrypting a file from Konqueror or Dolphin

Click on the file you want to encrypt with the right mouse button. Choose **Actions** → **Encrypt File** in the pop up menu. You will then be prompted with the Public key selection dialog. Choose

the key of the recipient and click **Encrypt**. The encrypted file will be saved with a `.asc` or `.gpg` extension depending on whether you chose **ASCII armored encryption** or not. ASCII encrypted files only use readable characters to represent the data resulting in files that are more robust when copied around or sent by mail but are one third larger.



### 3.3.2 Encrypting a text with KGpg's applet

You can encrypt the contents of the clipboard by selecting the **Encrypt clipboard** item in applet menu. When you choose **Sign clipboard** then the text will be signed instead. Both actions will import the current clipboard contents into an [editor window](#), perform the requested action and paste the contents back into the editor.

### 3.3.3 Encrypting text from KGpg's editor

This is as simple as clicking on the **Encrypt** button. You will then be prompted with the Public key selection dialog. Choose your key and click **Ok**. The encrypted message will appear in the editor window.

Usually you can only encrypt files with keys that are trusted by you. Since you sometimes want to just send a confident note to some random people you are aware of having a GPG key you can set the option **Allow encryption with untrusted keys**.

To make sure that you can decrypt every file you have encrypted even if they are encrypted with someone else's key you can use the options **Always encrypt with** and **Encrypt files with** which are available in the [KGpg configuration](#).

For more information on the encryption options **ASCII armor**, **Allow encryption with untrusted keys** and **Symmetrical encryption**, please refer to [gpg's documentation](#) or [man pages](#).

## 3.4 Decrypting Your Data

### 3.4.1 Decrypting a file from Konqueror or Dolphin

Left click on the file you want to decrypt. Enter your passphrase and it will be decrypted. You can also drag an encrypted text file and drop it into KGpg's editor window. It will then ask the passphrase and open the decrypted text in KGpg's editor. You can even drop remote files ! You can also use the **File** → **Decrypt File** and choose a file to decrypt.

### 3.4.2 Decrypting text with KGpg's applet

You can also decrypt the contents of the clipboard with the **Decrypt Clipboard** menu entry of the KGpg applet. An **editor window** will show up with the decrypted text.

### 3.4.3 Decrypting a text from the editor

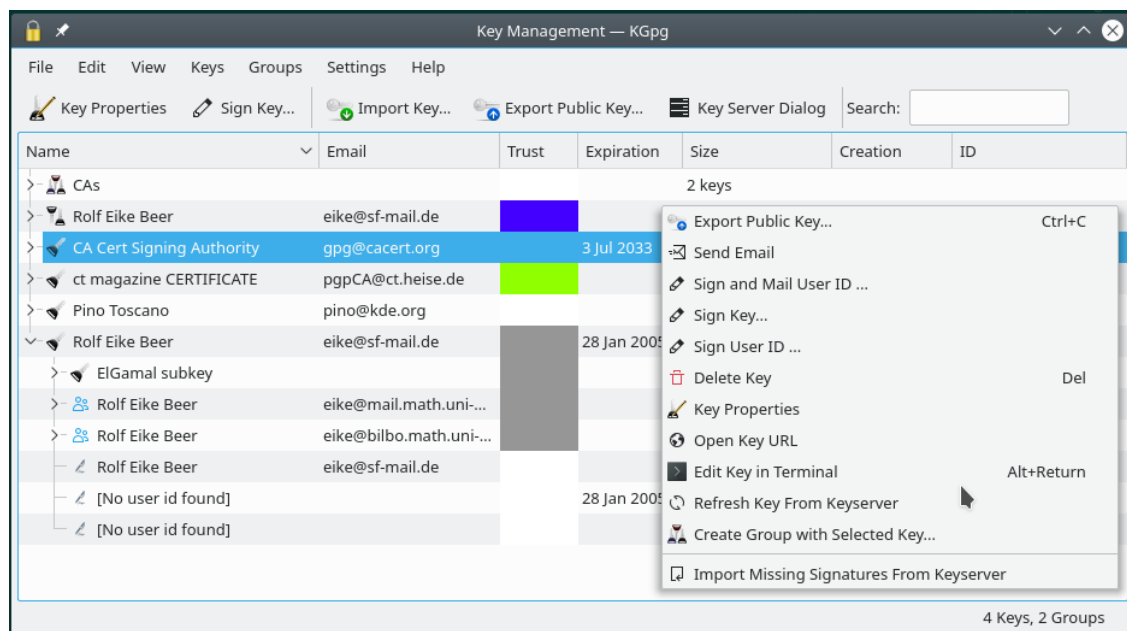
Copy or Drag and Drop the text you want to decrypt, and click on the **Decrypt** button. You will be prompted for the passphrase.

## 3.5 Key Management

All basic key management options can be performed through KGpg. To open the key management window click the left mouse button on KGpg's applet. Most options are available with a right click on a key. To import/export public keys, you can use drag and drop or the Copy/Paste keyboard shortcuts.

You can export a public key via email, to the clipboard, to a keyserver or to a local file. Use the options in the export dialog to export everything, export without attributes (photo ids) or export a clean key i.e. the key itself including its subkeys, but excluding all signatures.

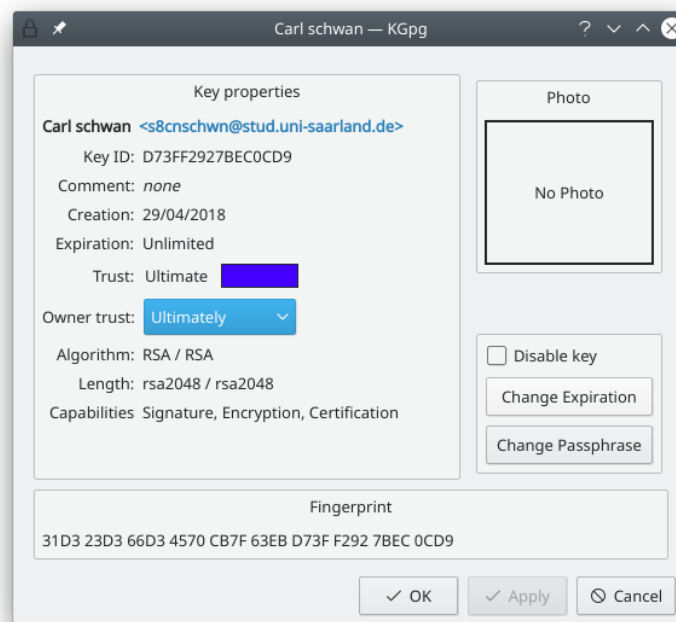
### 3.5.1 Key Manager



In this example you see a key group containing two keys, two key pairs and three public keys. The third column shows the trust you have in the keys. The first key pair is ultimately trusted and is also set as the default key (bold font) while the second one has expired. Two of the public keys are fully trusted while the trust of the last key is marginal. The last key is expanded, showing its ElGamal subkey, an additional user id, both also with marginal trust, and some of its signatures.

Signatures allow navigating through your keyring. Double clicking on a signature or a key shown as member of a group will jump directly to the corresponding primary key.

## 3.5.2 Key properties



While the key manager allows you to do general actions with one or multiple keys, key groups or signatures, the key properties window gives you access to a single key. You can reach it by pressing enter in the key manager or double clicking the key.

In this window you can change the key passphrase and expiration of your secret keys. For all keys you can also set the owner trust value.

This value indicates how much you trust the owner of this key to correctly verify the identity of the keys he signs. Taking the owner trust into account gpg creates your own web of trust. You trust the keys you signed. If you assign owner trust to these persons you will also trust the keys they have signed without the need that you first have to sign their keys too.

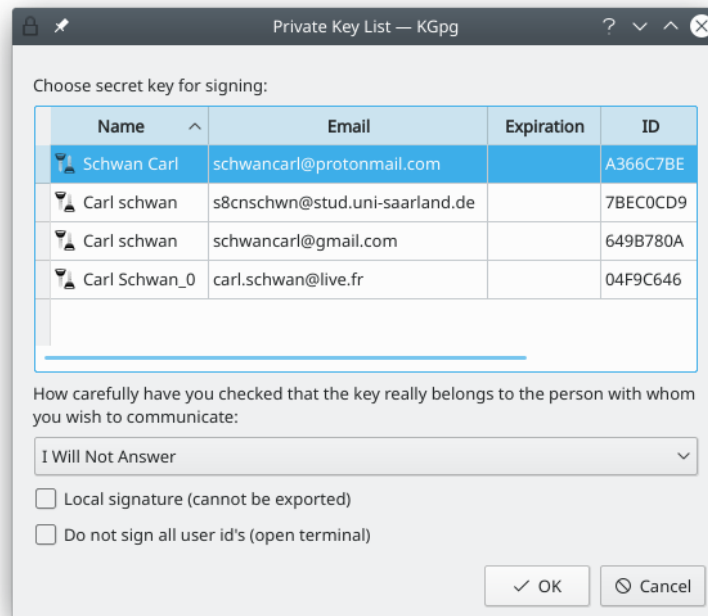
## 3.5.3 Signing keys

When you sign a key of someone else (let's call her Alice) you announce that you are sure that this key really belongs to that person and the key can be trusted. Of course you really should have checked that. This usually means that you have to meet Alice, check at least one identity card and get the full key fingerprint or a copy of her key. Then you go home and sign that key. Usually you will later upload the newly signed key to a [key server](#) so everyone knows you have checked that key and the owner may be trusted. Alice will likely do the same so you both will have your keys signed by the other one. If one of you has no identity card at hand it's no problem if the signing happens in only direction.

## The KGpg Handbook

But think about what happens if Alice lives on the other end of the world. You communicate with her regularly but there is no chance you will see her anytime soon. How do you trust her key?

When you select her key and then choose **Sign Key...** you will get the dialog that allows you to choose the options how you would like to sign that key.



First you can choose the key you will use to sign the key. Then you can enter how carefully you checked that she really is the person she pretends to be. This information will be stored together with the signature so it is a guidance for everyone else who might need that signature (more on this below). And then comes the option that would help you if you can't meet Alice in person: **Local signature (cannot be exported)**. When you activate that option a special version of a signature will be created that can never even by accident leave you keyring.

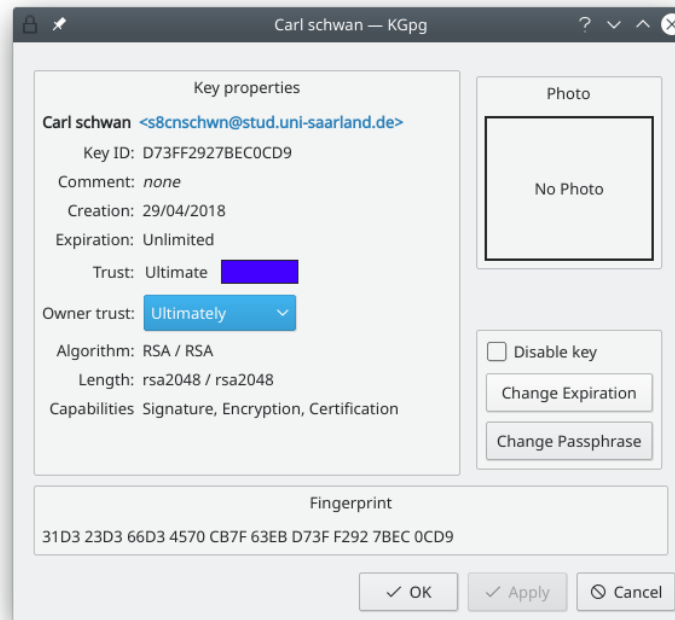
But why is it important how carefully you checked Alice's identity? Who should care? There is a different way to solve your problem with the identity of Alice. If you can't visit Alice anytime soon just think of Trent. You know Trent has a keypair, too. And Trent is a globetrotter, being on a different continent at least twice a month. If you are lucky he will fly close to Alice soon. So you will go and meet with Trent to sign keys. Then you will drop Alice a note that Trent will be at her place soon and ask her if she can meet with him too to sign keys. After all this has happened you know that Trent's key can be trusted and Trent knows that Alice's key can be trusted. If you trust Trent that he has carefully checked Alice's identity then you can also trust her key.

These relationships between keys and their owners form a so called web of trust. Within that web there are some important values that define how trustworthy a particular key is. The first thing is how carefully the identity of the key owner was checked. That is the value you have seen above in the secret key selection window. For example you will likely know how to verify your local countries identity card but one from a completely different country may be hard to verify. So you could say that you have very carefully checked Trent's identity because you have seen his identity card and it looks very much the same as yours. But Trent, although he has seen both Alice's identity card and driver license might say he has only done casual checking of her identity as he is not absolutely sure about the documents from that part of the world.

The next important value is how much you trust the other person to verify documents. You know Trent is good at that. But George for example is no one you would call smart. He barely looked at your id card when you met him for key signing. You are sure that George is the person he pretends to be as you checked his documents carefully. But he doesn't seem to really care if he

## The KGpg Handbook

checks other people so you will have a high trust in the key of George but a very low trust in the signatures of George. If you open the [properties](#) of a key you will find the field **Owner Trust**. This is how much you trust the key owner when he signs keys. This value will not be exported, it is completely up to your personal preference.



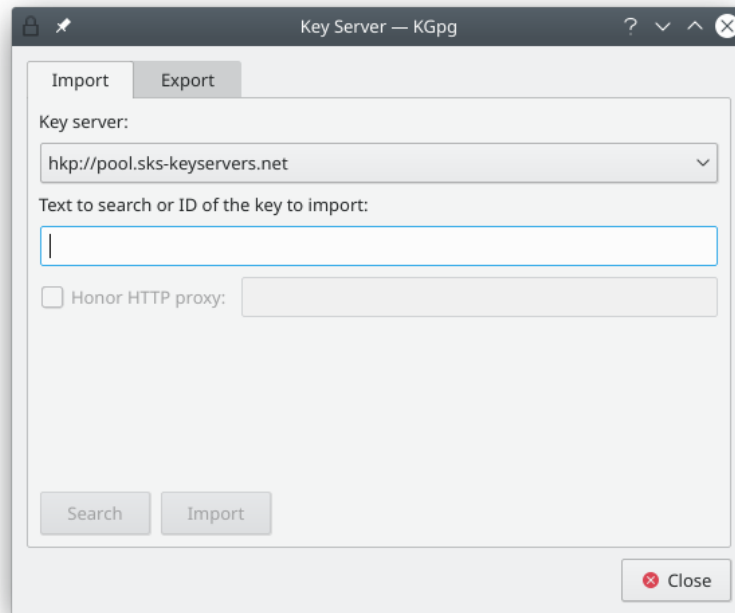
Now you should have an idea how the web of trust is built, what the owner and key trust values are for, and why you always should be very careful when checking identities: other people might rely on you. But one element in the process is still unverified: the email addresses in the keys you signed. Creating a new user identity in your key with the email address of Alice or Trent will only take a few mouse clicks. You have verified that Trent really owns his key. But no one has checked until now that Trent really controls the email addresses of his user identities.

If you choose **Sign and Mail User ID...** from the menu instead you can close that gap. The idea is that you will sign the key as usual and afterwards it will be split into pieces. Every piece will only contain one user identity of Trent's key and your signature to it. This will be encrypted with Trent's key and sent only to the email address given in that identity. Only if Trent can receive this mail and decrypt the message he will be able to import that signature into his key ring. You will not upload your signatures, this is entirely up to him. If your signature will show up on a key server you can be sure that Trent really controls both his key as well as the email address you signed. The signatures you make in this process will also be not part of your keyring. So right after you signed Trent's key it will still be shown as untrusted in your keyring. Once Trent has received your mail and imported your signature into his keyring he can upload them to a keyserver. When you refresh his key from a keyserver you will get the new signatures. While that may sound inconvenient first it makes sure that you will not by accident see one of his identities as trusted that he does not control. Only the signatures that show up on a keyserver are those where everyone, including you, can be sure that he really controls the corresponding email addresses.

## 3.6 Working with key servers

### 3.6.1 Communication with key servers

The public part of a key pair is usually stored on a key server. These servers allow anyone to search for a key belonging to a specific person or mail address. The signatures are also stored on these servers.

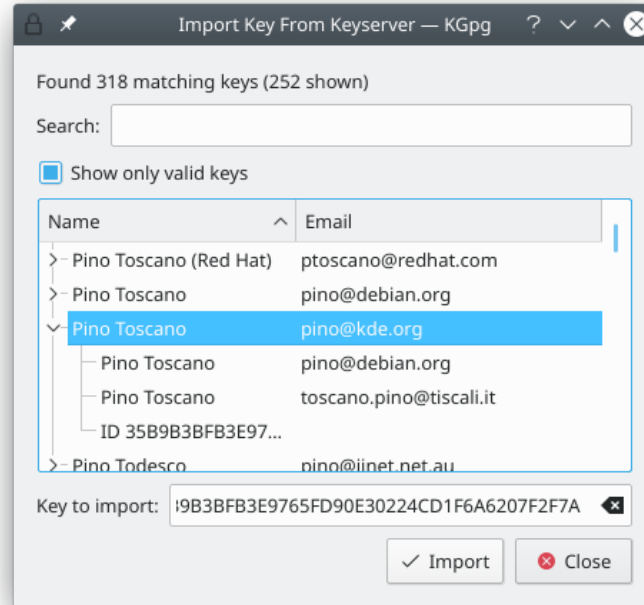


This dialog gives you access to the key servers. You can search and import keys from a keyserver as well as export keys to a server. An example of searching and importing is when you want to write a mail to someone new. If you would like to encrypt your mail to your contact you can search if he or she has a public key on the key servers. If you have created a new key pair or have signed someone else's key you might want to export the public key (possibly with new signatures) to a keyserver.

Most keyservers synchronize their data between each others so you will get similar search results regardless which server you use. Since there are exceptions of this rule you can choose the keyserver to use in this dialog. It's usually a good idea to choose a default keyserver that is located close to you (i.e. in your country or on your continent) as they usually respond faster to your queries.

Please note that everything you upload to a keyserver usually stays there forever. This is one reason you should usually limit the lifetime of your keys. Also note that the keyservers are sometimes scanned by spammers for email addresses.

### 3.6.2 Key server search results



All results of a search are displayed in this window. This picture shows a search for “@kde.org” addresses which showed up 244 results. Using the search field the displayed list was reduced to a single key. This key has two matches: the primary user id itself matches the search string as well as one of the other user ids.

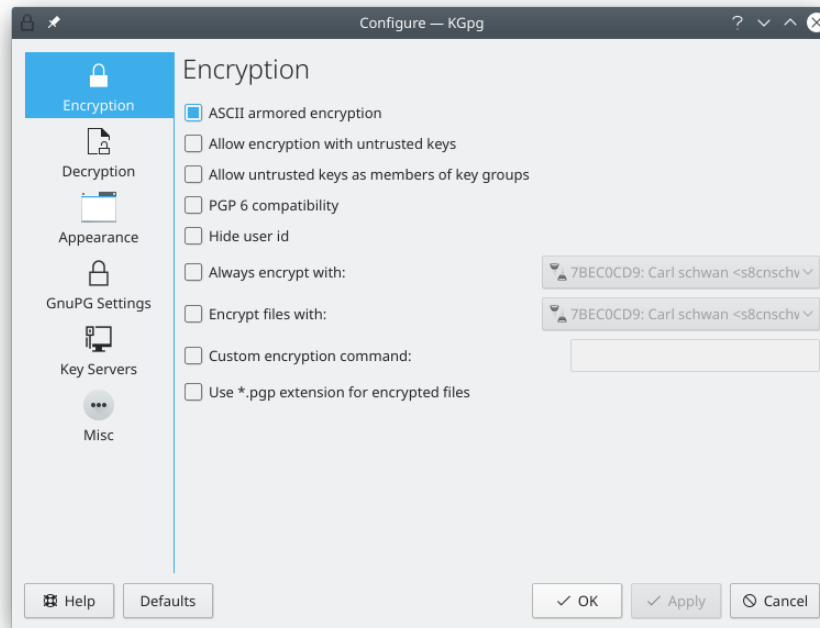
You can select one or more keys to import. The ids of those keys are shown in the **Keys to import** field at the bottom of the window. When you click on **Import** the key server is contacted again and the keys are fetched into your keyring.

## 3.7 Configuring KGpg

Configuration is accessible through the KGpg applet menu (right mouse button click on the applet) or through the main menu ( **Settings** → **Configure KGpg** ). You can set default parameters for encryption, decryption, user interface and applet. Most encryption options are directly related to gpg and are documented in it’s [man page](#).



### 3.7.1 Encryption



Here you can configure special options to be passed to GnuPG to change the encryption behavior. For detailed description please have a look at the GnuPG manual.

- **ASCII armored encryption:** this causes encrypted files to be stored in a format that uses only printable ASCII characters and has short lines. Files stored this way are bigger than the files in binary format but are easier to send e.g. by email.
- **Allow encryption with untrusted keys:** this allows you to encrypt files with keys that are not trusted by you.
- **PGP 6 compatibility:** encrypted files are compatible with the older PGP6 standard. This disables certain features so you should only use this if really needed.
- **Hide user id:** this removes all evidence of the receiver from the encrypted file. In case the transmission is intercepted noone could gain information about the recipient from the file. If the receiver has multiple keys he needs to try which one was used.
- **Always encrypt with:** all encryptions are additionally encrypted with this key. If you set this to one of your private keys this makes sure you can read all data you encrypted by the price of bigger messages.
- **Encrypt files with:** behaves like **Always encrypt with** for file encryption.
- **Custom encryption command:** if you need to pass some unusual options to GnuPG you can specify the command line here. Most users will not need this.
- **Use \*.pgp extension for encrypted files:** if you check this option encrypted files will be named as the input file with the extension `.pgp` added, otherwise the extension `.gpg` is used.

### 3.7.2 Decryption

Here you can specify a custom decryption command. This option is seldomly needed and only useful for advanced users that know of GnuPGs command line options.

### 3.7.3 Appearance

Here you can configure the way KGpg looks to you. Possible settings are the colors that reflect the different levels of key trust in the [key manager](#) and the font settings for the [editor](#).

### 3.7.4 GnuPG Settings

Here you can configure which gpg binary and which **configuration file** and home folder are used. These values are autodetected on first start and should already work.

Using the [GnuPG agent](#) makes work with GnuPG more comfortable as you do not need to type in your password for every action. It is cached in memory for a while so any operation that would require a password can immediately be done. Note that this may allow other people to use your private keys if you leave your session accessible to them.

### 3.7.5 Key Servers

Here you can create a list of key servers that are shown to you when you open the [key server dialog](#). If you run GnuPG from the command line only the key server you set as default here will be used.

The protocol used for communication with the key servers is based on HTTP, so it makes sense in some environments to **honor the HTTP proxy when available**.

### 3.7.6 Misc

This section allows the setting of some different features that do not fit into the other sections. You can configure for example to **start KGpg automatically at login**. The option **use mouse selection instead of clipboard** changes if selection happens by mouse and pasting by middle mouse button or if all operations are done by keyboard shortcuts.

You can also change if the systray icon of KGpg is shown or not and what action happens if the icon is clicked with the left mouse button. If the systray icon is shown closing the KGpg window will minimize the application to tray. If the systray icon is not shown KGpg will exit when all windows are closed.

## Chapter 4

# Credits and License

KGpg

Program copyright (c) 2002-2003 Jean-Baptiste Mardelle [bj@altern.org](mailto:bj@altern.org).

(c) 2006-2007 Jimmy Gilles [jimmygilles@gmail.com](mailto:jimmygilles@gmail.com)

(c) 2006,2007,2008,2009,2010 Rolf Eike Beer [kde@opensource.sf-tec.de](mailto:kde@opensource.sf-tec.de)

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).