K-tools: SIMPLE SECURITY

Keeping your emails secure has always been a fairly tricky business but Stefanie Teufel has the secret to security success

Keeping a secret to oneself is not always easy – especially in mail traffic. Although many people would prefer not to think about it, emails are comparable to a postcard. Every intermediate station the mail passes can read the message if it really wants to. Maybe that hadn't occurred to you?

What a good thing there's Geheimnis (it means "secret" in German). With the aid of this KDE front-end for the most common encryption programs, all your dark secrets will be hidden from even the most curious virtual postmen. You are even secure from snooping colleagues, because in addition to mails, the program can also be used to make child's play of encrypting important files on your hard disk.

To keep your mails and data secure in future, you now need nothing more than the latest release of the tool, which you will find on the homepage of the authors Chris Wiegand and Stefan Suchi at *http://geheimnis.sourceforge.net/*, plus a functional encryption program. In the case of the latter, by the way, you have more or less a free choice, since Geheimnis gets along with **PGP** 7.0, PGP 6.5, PGP 5.0, PGP 2.6.x and also **GnuPG** 1.0.x. Even using different versions and programs at the same time is no problem, since the tool allows different profiles to be created.

SuSE users are especially lucky, since there is even an rpm package available for them to download. The owners of other distributions, unfortunately, will have to get their hands, or rather their compiler, dirty and install Geheimis after unpacking the sources (*tar –xzvf geheimnis-1.96.tar.gz*) and then:

```
./configure
make
make install
```

You should also carry out this Linux three-step in the geheimniskeepopen subdirectory which is created when you unpack. Please do not forget this, because Geheimnis needs this auxiliary program. To be specific, this is a simple **Wrapper** around a shell, which keeps the shell window open until the user explicitly closes it. This is necessary so that one can read the outputs of the active command line PGP program during all actions.

K-tools

In this column we present tools, month by month, which have proven to be especially useful when working under KDE, solve a problem which otherwise is deliberately ignored, or are just some of the nicer things in life, which – once discovered – you wouldn't want to do without.

Encrypted

If you can hardly wait to finally make your data secure, you should waste no time in starting the program. To do this, enter either a simple *geheimnis &* in any terminal of your choice. Anyone who is familiar with the program from the good old days of KDE-1.x will be a little disappointed that Geheimnis in the current version 1.96 cannot yet dock in the panel. So a search for the old familiar menu item *Applications/Geheimnis Dock Menu* at present will sadly be in vain. The back-story to this is that the mechanism for docking has changed completely between KDE 1 and KDE 2, so the corresponding code will have to be completely rewritten. But don't worry – according to the developers, the re-implementation of this practical feature is right at the top of their To-do list.

PGP This abbreviation stands for Philip Zimmermann's encryption and decryption program *P*retty *G*ood *P*rivacy". From the point of view of the user, PGP works as a "Public Key System", which means that two different keys, which go together are used. Number one is the public key, which the user is supposed to circulate among the population as generously as possible, while Number Two is a secret key, which must only be in the possession of an authorised person, meaning you. With the public key of a third party, one codes messages to the latter in such a way that only the holder of the related secret key (which is the third party) can then decrypt them.

GnuPG A complete and Free substitute for PGP. Unlike PGP, GnuPG is Free software, which means the program's source code is freely available, free from patents and free from restrictive licence conditions.

Wrapper Wrappers are a type of software which encapsulates an object in such a way that it is easier and/or more secure when in use than the original, unwrapped object. An object in this case can be, for example, a program or a protocol.

Geheimnis greets you in the first instance by telling you that it's time to create a new profile, preferably a separate one for each encryption program you use (Figure 1).

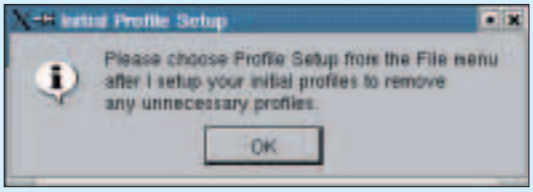

Figure 1: Show your profile!

Click on the OK button to start. The next window (Figure 2) is also an advisory screen, in which you are prompted to enter the path to the subprogram geheimniskeepopen. Normally the program is found under */usr/local/bin*. If you are unable to track it down there, a **which** *geheimniskeepopen*, or if necessary a **locate** *geheimniskeepopen* will help you.

Enter the result in the window from Figure 3, which your new PGP tool opens for you automatically. Then tell Geheimnis which encryption program you would like to use. To do this, click in the section *Profiles* on the *New* button. In the window which then appears (Figure 4), you can seek out the appropriate program in a pull-down menu and give the baby a name in the box underneath. The name entered there appears, by the way, later in the profile selection.

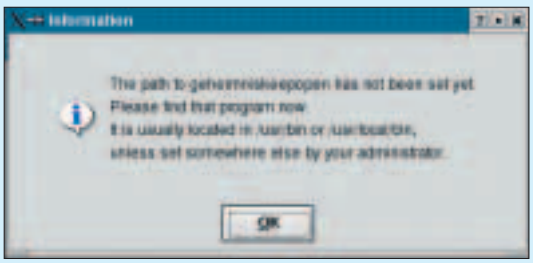

Figure 2: Are you on the right path?

which and locate With the *locate* command, you search for files in your filesystem. The command does this by accessing a database (*/var/lib/locatedb*), which, with the command *updatedb*, is created (or updated). A list of all the matching files with full path specification is output. The *which* command on the other hand searches all the directories in the specified path for the specified command.

Sign By signing a (public) key you are ensuring that this key does actually belong to the person whose name is on it. You should therefore only sign such keys as you have received personally on diskette from its owner, who is known to you, or for which the owner has personally provided you with the fingerprint e.g. by telephone. This "Fingerprint" is a series of characters generated from the key data, which marks the key unequivocally, but does not allow any information to be gathered about the key itself. When you sign a text or a file, PGP leaves the original data in clear text and attaches to it a signature also created from the text data (in 7-bit format for sending emails). When the receiver forms a test fingerprint from the data received with the same algorithm, he can then check if the data have reached him unaltered.

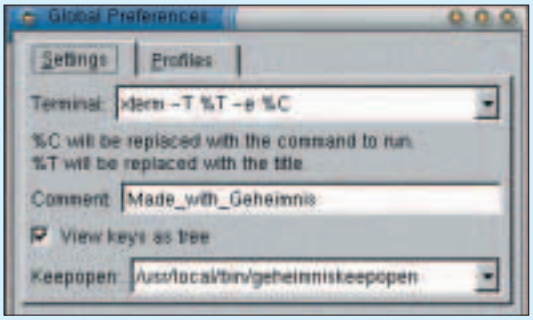

Figure 3: Nice and clear, the secret central control

The lord of the rings

It's now time to think about the key. Click in the starting window (Figure 5) on menu item Key Management. Geheimnis then reads in – if you are already in possession of a key ring – all the necessary files and presents you with the result in a window as in Figure 6. Once loaded, you can potter about to your heart's content with your keys. Move the cursor onto the key concerned, press the right mouse button, and decide for yourself if you want to delete, **sign** or otherwise process the key.

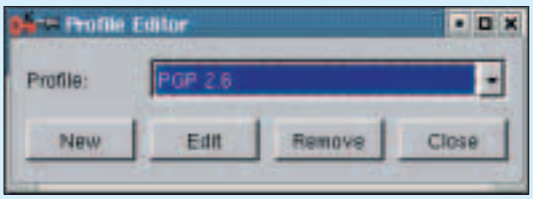

Figure 4: Which encryption program are you going to use?

You can easily recognise your own, private key pair by the fact that firstly it is blue and secondly it consists of two keys. Keys marked in grey signal that these have been withdrawn by the owner (*disabled*), so should no longer be used. The standard setting is red. This colour tells you that you do not classify the key as verifiably genuine. Verified keys can be recognised, on the other hand, by their green colour.

PGP newbies select the menu item *Key management/Create key pair*. In a terminal window which opens automatically you can now use geheimniskeepopen to track, with no worries, how Geheimnis executes the command which is necessary to generate a key, *pgp –kg*. Since PGP is usually highly talkative, you then only have to follow the instructions, to end up with your own personal key pair.

It is just as simple as key management to encrypt and decrypt files. Drag the file to be encrypted out of

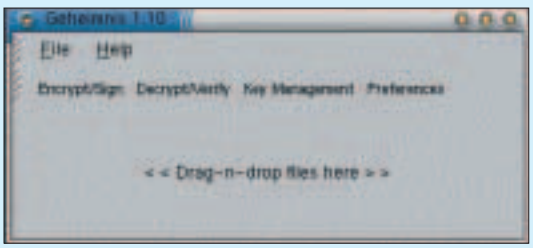

Figure 5: The main window

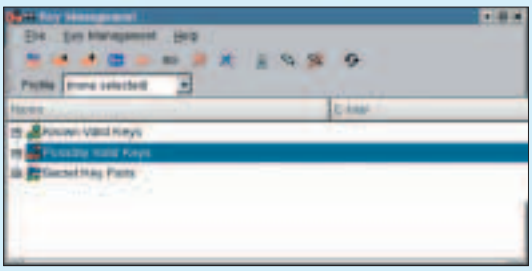

Figure 6: Your virtual bunch of keys

Konqueror using drag and drop to the box labelled "Drag-n-drop files here" in the main window from Figure 5.

This is where you then choose the box Encrypt/sign file. In the next window (Figure 7) you should then specify the appropriate profile and choose which public key should be used. Instead of encryption it is also possible to merely sign the file with your key. If you do this, people who are not working with PGP can also read the corresponding message. But in this case, bear in mind that they cannot check your signature for authenticity.

If you do not wish to pass on a file, but simply keep it encrypted on your hard disk, you don't need to select a second key. In this case, activate the field Encrypt for self.

If you don't like the file name suggested by Geheimnis, simply change it. It is also possible to rename it later without any problem. But do take care to retain, if possible, at least the ending *.pgp*, since this is how the file remains linked to the program Geheimnis. In this way you are making sure that your PGP front-end will later automatically be started for decryption when you click on the file in *Konqueror*. That's how easy cryptography can be.

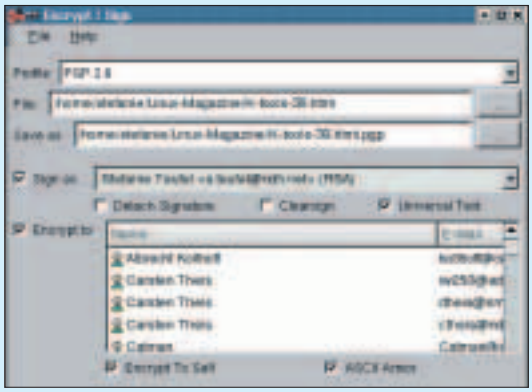

Figure 7: It's entirely up to you who can decrypt the file in future