# Encryption – Part II

## *By Andy Pepperdine*

Encryption is the name for methods of hiding information from some, while allowing others access to it.

## *Introduction*

This part will describe how you can protect your data by encryption in individual files, in a special folder, or in a file system embedded either in an ordinary file or on a partition on, for example, a portable memory device (like a USB memory stick). It is assumed that you have read the previous part to see the basics.

## *Single files*

This technique can use the same secret key as signing and encrypting e-mail messages described in part I.

There is in Ubuntu a package called Gnu Privacy Assistant (GPA), a graphical front end for the GnuPG package which can encrypt single files as required. It is found by default under Applications → Accessories.

Then, from Windows → Filemanager, or the Files icon and select Open, a file browser opens, whereby you can navigate to the file you wish to encrypt. Select (highlight) the filename you wish to encrypt and hit OK. Now you can select the operation to be performed. To encrypt, choose Encrypt and from the list of keys, select the one you need.

**Remember:** you can encrypt using any public key, but decryption requires you to have access to the secret key (see part 1 of this presentation, last month). To keep a file private to yourself, make sure you use one of your own keys.

This creates a new file with file extension .gpg to indicate it is encrypted.

However, note that the new file will have the modification date as the true date it was created, and NOT the same as the original file. On Unix/Linux systems, there is a useful command that can set the timestamp on a file to the same as that of another file. After you have encrypted, it would then be possible to do this on a command line:

```
touch -r oldfile oldfile.gpg
```

which will set the new file's dates the same as the old one.

Decryption is done similarly, but again you may have to copy the dates and times across.

Alternatively, you can use the file browser (Nautilus) to navigate to and select the file you wish to encrypt, and then go to Edit → Encrypt, when it will present you with the Passwords manager to say which key you want to encrypt with. It selects all the public keys, so make sure it is for the right recipient who will be the only one who can decrypt it. Note that selecting the key is not enough, you must also check the box at the start of the line containing the key to use. Then the OK button will cease to be greyed out. But, again, however, it does not set the date and time as for the old file.

Double clicking an encrypted file in Nautilus will automatically decrypt it if possible.

## *Special cases of files*

Some files can be created protected by passwords. This will use the encryption technique defined for that file type and the strength of the protection will depend on the definition of the way it is to be done. Microsoft has their own method for .doc files, but the .odt standard uses methods are are defined by the standard and are available to every application that can read and create these files (like OpenOffice, Kword, Abiword, etc.). When saving a file in one of these formats, the dialog box should show some means of specifying a password to enable the protection.

Another type of file that has this feature is the .pdf file. It also is standardised, and OpenOffice gives the option in the Security tab of the File → Export to PDF menu, where it is called Encryption with a check box and a dialog to set the password.

## *Special cases of folders and collections*

The .zip format of files also has the option of password protection. On Ubuntu, this is accessible from the file browser by selecting the items ot be collected together (or single folder) and then Edit → Create Archive. Select the file type of .zip and click the button to expose Other Options. You can then encrypt with a password either just the files, or both files and the names of files. The .zip format is universally known so the files can be transported to their destination without further work.

## *Folders of private files*

Encrypting a single file is all very well for occasional use and for copying single files to a stick to take away, but it is not a practical way to keep all your secret files secret automatically and without needing to explicitly encrypt them yourself.

Ubuntu (Intrepid Ibex) comes with a method of defining a folder which can contain just such a set of encrypted files, protected by a password when you first want to use them.

First, install the package ecryptfs-utils.

To create the special folder to hold the secret files, you have to use the command line:

```
ecryptfs-setup-private
```

and log out and in again to mount the special folder. You are given an opportunity to create a “mount passphrase”, and this it is that you will to recover the data from outside your login account.

You will now find that you have two new folders in your home directory, named `Private` to give you access to your secret files, and `.Private` to contain the real files and is normally hidden. When you browse to the Private folder the first time, you will be presented with a request to save the encryption passphrase which is usually automatically used when you log in, but which must be used specially otherwise. Generate this according to the instructions and save it safely.

If you can still login then the passphrase can be recovered by means of the command line:

```
ecryptfs-unwrap-passphrase
```

Do not operate on the files in the hidden folder directly. They are encrypted and you could mess up the decryption, especially as the names of the files are also encrypted from Ubuntu Intrepid onwards.

**Note:** if you have auto-login for your account, then the Private directory will NOT be automatically mounted for decrypted access. This is deliberate in order to force a password to be entered at least once on the machine before access is allowed. You can gain access by looking at Private in Nautilus and there you will see two files. Double clicking on Access-Your-Private-Data.desktop will prompt for the mount passphrase (not the same as your login password) and mount the private data.

The downside of this is that the files are not portable as they are tied to the user's installation, although the whole directory can be moved to another similar Linux system.

## *Portable filesystems*

If you wish to put your encrypted files onto a portable memory device (like a USB stick), then the best option by almost all accounts is the use of free software called Truecrypt [Tru]. This can support all modern versions of Windows, Apple machines, and Linux (Suse and Ubuntu easily, the others via source compilation) and so can give confidence that you can extract the data afterwards on some other machine.

Insrtallation on Windows is described at [Edi]. As the download is an installation executable, it could be put also onto the portable device so that you can install it at the recovery site if it is a Windows system.

Installation for Linux is a matter of downloading the compressed tar-file and then unpacking it. Running the resulting shell script gives you the option either of installing directly, or of extracting the .deb file (Ubuntu). The latter method is the way I did it, in order that I can keep the .deb file for later use if need be.

Installing the package is nothing more than double clicking on the .deb file in Nautilus in Gnome, checking the status and hitting the Install Package button. After installation, it appears under Applications → Other.

## *Basic notions*

Truecrypt can create encrypted file systems in several different ways. I'll describe two of them.

First, it can allocate and use a normal file in any file system (e.g. NTFS), and build a pseudo file system inside it. So far as anyone else looking at the file is concerned it looks like a file full of random data. There is no indication that it contains any accessible information unless a password is given to access it. When the magic words are given, then the file turns into a full file system looking like any other.

Second, it can encrypt a whole partition on an external device. I will assume that we will want to encrypt a data partition. Encrypting a system partition is possible for a Windows machine, but needs more careful set-up which I will leave the reader to investigate should they wish to render their whole machine useless to anyone without the password to boot it (and you if you lose the key!).

## *Simple case*

When you start Truecrypt, it shows what it knows about. To create a container for the encrypted file system, the easy option is to create a normal file to act as the container. So press the Create Volume button.

The next question is defaulted to build a container in a file. Hit Next.

The next question is regarding hidden volumes, that will not be covered in this paper, but it is very clearly described in the Truecrypt documentation on their website and also why you might wish to hide an encrypted file system.

Continue with the dialog in an obvious fashion until it finishes with a Format button.

Back in the Truecrypt window, you can now select an empty slot, browse to the file just created, and mount it. An icon appears on the desktop and the system believes that there is a new file system. When it is mounted DO NOT manipulate the file in any way directly. Unmount before you try to copy, move or delete it.

NOTE 1: All mounting and unmounting (dismounting in Truecrypt parlance) should be done through the Truecrypt interface.

NOTE 2: If you want to transfer encrypted files between systems, then you will have to choose FAT as the file system, otherwise it will warn you that you will have to install extra drivers on the target systems too.

When it is unmounted (and not before!), the file can be transferred anywhere in any manner, but it will look like a single file full of random data. Any machine equipped with Truecrypt and the right password can then unlock the file and make it look like any other set of folders.

## *Encrypted partition*

Here we'll look at creating an encrypted file system in the whole of a partition on a USB device, like a memory stick.

When you plug in a USB stick, it will show up on the desktop if it can be mounted. To use the whole device as an encrypted one, you must unmount; and then reformat it using Truecrypt.

NOTE: When creating this type of partition, any data on the partition will be lost permanently, so make sure you know what you are doing.

Opening Truecrypt and hitting Create Volume as before starts the process. This time, select Create a Volume within Partition, and on the next window select Standard Volume.

When selecting the device, it will show you all the devices that can be seen. Those which are already mounted will have their mount points also displayed. Select the USB device you plugged in, making sure that the mount point column is empty.

It then warns against doing this if you are new to encryption, as it renders the whole partition unavailable rather than just a single file, and a stranger may not understand what is going on and re-format it again if prompted by their system. Ignore this and proceed.

Carry on in the obvious way until the formatting is complete.

To use the device, Mount and select the it, and it will appear on the desktop as a removable device, although it can only be unmounted via Truecrypt and not directly via the desktop.

When done with, unmount with Truetype and remove it.

When plugging it in again, the partition will not appear in the attached devices, but starting Truecrypt will list it when selecting the device. Mounting that device will challenge for the password and then give you access to the files on it in the usual manner.

## *Separating encrypted from non-encrypted data*

If you want to put on the same memory stick both encrypted and plain data, then it is best to create the partitions you require on the device before you start. On Ubuntu you can do this by installing the package *gparted* and looking at the device after it has been unmounted. gparted is placed under System → Administration → Partition Editor, and gives you the option of navigating to the correct device. When you have edited the new state of the partitions as you require, then you must hit Apply before they will be acted upon.

NOTE: When you do this, you will find on Ubuntu, that the new partitioning is not visible by the GUI on the desktop until the device has been unplugged and re-plugged in. Then it all springs to life.

In this way, you can store on one partition your normal data, including the Truecrypt software that could be installed at the target machine to receive the encrypted data, and a second partition can be the encrypted partition.

## Warning

Some USB memory devices do not behave as I expected. For instance, I now have one which appears to a PC as thogh there are two separate devices. The smaller of these can only formatted as a Unix filesystem (e.g. ext2) and Windows can no longer format it to VFAT. The other partition, I eventually formatted as an ext2, and subsequently it then took a FAT32, making it once again useful for general purpose transfer of data. I am not sure what it was exactly that messed up the device.

I have had no problems at all with those under Maplin's own name – they behave just like you would expect a disk drive to behave.

## *References*

[Edi] *http://staff.napier.ac.uk/NR/rdonlyres/0BF42486-78F5-4971-85BE-5D2F4534FC55/7855/STAFFTruecryptInstallationGuide2.pdf* , is a useful recent PDF file describing how to install on Windows. Omit the download section if you keep the file on the portable device.

[Tru] *http://www.truecrypt.org/* is the home page of the developers of Truecrypt from where you can also get a lot of

[Ubu] *https://help.ubuntu.com/community/EncryptedPrivateDirectory*, contains a “how to” covering all the details of how to set up and use the Private directory.