

The encrypted dual boot laptop howto

(or 'The encrypted dual boot single hard drive system howto')

(or 'The encrypted root openSuSE 10.3 howto')

Ed 'Shappy' Hopper
(<http://www.shappyhopper.co.uk>)
October 2007

Synopsis

This howto is aimed at those people who want to encrypt a dual boot Windows XP/ Linux (openSuSE 10.3 is used as the example) system that resides on one physical hard drive using only free software.

Adaptations (using only certain sections) of this howto include encrypting an opensuse 10.3 installation and encrypting/ securing a 64 bit Windows installation.

At the end of this howto the user will have a Windows system with an unencrypted Windows C: drive (with free-space wipe available) but encrypted pagefile and user settings, making it very unlikely that any secure data should ever 'leak' out.

The user will also have a fully encrypted Linux system, including encrypted swap and encrypted hibernation.

Both systems will share a fully encrypted Ext3 partition containing /home and the users 'Documents and Settings' folder. 'My documents' can be shared between Windows and Linux on this system.

This method can be adapted for Windows Vista systems, but the user should be aware that very few Vista systems come with the necessary software to allow a complete re-install.

Contents

1	Introduction	5
2	Overview	6
2.1	The final system – dual boot	6
2.2	The final system – 64 bit Windows	6
2.3	The final system – openSuSE 10.3	6
2.4	Will my system be fully secure?	6
2.5	How could information leak onto the unencrypted Windows Partition? 7	
3	Setting up	8
3.1	Getting started	8
3.1.1	Checking the downloads	8
3.1.2	Determine C – drive required size	9
3.2	Wiping all data from the HDD	10
4	Windows – Part 1	12
4.1	Installing Windows	12
4.1.1	User accounts	12
4.1.2	Disable LM Hashing	13
4.1.3	Get system up and running	13
4.1.4	Check the pagefile size	13
4.2	Install Software	14
4.3	Single OS Windows System	15
5	Linux – Part 1	16
5.1	Installing openSuSE 10.3/ Linux	16
5.1.1	Partitioning	16
5.1.2	Grub	17
5.1.3	Software	17
5.1.4	Update	18
5.1.5	Root password and user accounts	18
5.2	Randomising the root partition	18
5.3	Setting up the bootloader - Grub	20
5.4	Encrypting and formatting root	20
5.4.1	Making a cryptab file	20
5.5	Moving the root Filesystem	21
5.6	Mount options: fstab	21
5.7	The initial Ram Disk - Initrd	22
5.8	The first boot, what to do if it doesn't work, and what to do when it does	22
5.9	Creating the /home directory	24
5.9.1	Installing truecrypt	24
5.9.2	A note about Truecrypt and new kernel updates	24
5.9.3	Remove old items from grub	25
5.9.4	Creating the home directory	25
5.9.5	Formatting the home directory	25
5.9.6	Setting up an auto-mount for /home	25
5.10	Adding a user account	26
6	Windows - Part 2	27
6.1	Installing TCGINA and encrypting the Windows user account	27
6.2	Moving the Windows pagefile to the Linux Swap partition	28

6.3	Wiping the free space on Windows.....	28
7	Encrypting the system swap and hibernation space	29
7.1	Swap Space	29
7.2	Hibernation	29
7.3	Simple steps – Wiping/ encrypting Swap and deleting/ encrypting the pagefile.....	30
7.3.1	Linux – wipe the swap partition	30
7.3.2	Windows - Delete the pagefile.....	30
7.3.3	Linux – simple encrypted swap without hibernate or Windows pagefile share	31
7.3.4	Windows – Encrypting the pagefile on the fly	31
7.3.5	Linux – Encrypt the hibernate disk image.....	32
7.4	Expert steps – Encrypted pagefile, swap and Linux Hibernate.....	33
7.4.1	Warning	33
7.4.2	A note on Windows hibernate.....	33
7.4.3	Boot manager – use Grub	34
7.4.4	Creating an encrypted swap system, compatible with pagefile sharing and using encrypted Linux hibernate	34
7.4.4.1	Getting started.....	35
7.4.4.2	Remove swap devices from /etc/fstab	35
7.4.4.3	Edit /etc/init.d/boot.local	35
7.4.4.4	Edit /etc/init.d/halt.local.....	35
7.4.4.5	Create a script for hibernating the computer	35
7.4.4.6	Filling the swap partition with random data	36
8	Bugs and issues to note	37
8.1	Unable to mount Truecrypt in Windows after Linux crash and vice-versa	37
8.2	Resume fails or data is missing after hibernate	37
8.3	Spyware software doesn't work	37
A.	Dummy initrd script	38
B.	Grub entry	39
C.	Cryptab Entry Example	40
D.	Root FS move command.....	40
E.	Fstab example	41
F.	/etc/init.d/boot.local example - Truecrypt	42
G.	Example /etc/init.d/halt.local.....	43
H.	Example hibernation script file	44
I.	Example SwapFs Registry Entry	45
J.	/etc/suspend.conf example	46

1 Introduction

The reasons for whole hard disk encryption, especially on laptops, have been covered elsewhere on the internet and so they will not be repeated here.

What normally stops people from encrypting their laptop is the cost of encryption software, and that is where this guide comes in; the encryption tools described here are all free.

Most home PC's and laptops have only one hard disk. For exclusive Windows XP and Vista 32 bit (with the exception of Vista Ultimate which comes with its own optional HDD encryption) users who want to encrypt their whole hard drive for free there is CompuSec:

- (http://www.ce-infosys.com/english/downloads/free_compusec/index.html)

Sadly this package doesn't work with a dual boot system on a shared hard disk (or 64 bit versions of Windows) because it encrypts the Linux bit as well as the Windows bit¹.

This guide is for everyone else; those who have a dual boot Windows/ Linux system on a single hard disk (e.g. on a Laptop), or who use a 64 bit version of Windows and want to secure their computer for free.

I have tried to make this howto as simple as possible, but have assumed that the reader will have a level of familiarity with installing Windows and Linux.

¹ Note: CompuSec do provide Linux drivers, but these are pre-compiled and only suitable for a few distributions. Also there have been no updates to the Linux version since early 2006.

2 Overview

This howto is for people using both 64 bit and 32 bit Windows XP (Professional and Home) and openSuSE 10.3. This can (I hope) be easily adapted for Vista and other Linux system users.

The following section describes the final system overview depending on options followed.

2.1 *The final system – dual boot*

Once completed your system will have:

- An unencrypted Windows partition with no user data or pagefile
- An encrypted² swap partition used by both Windows (pagefile) and Linux
- An encrypted openSuSE 10.3 root
- An encrypted documents partition shared between Windows and Linux containing all user information from both systems

A shared home/ documents partition frees up a significant amount of disk space when compared with a standard dual boot installation.

2.2 *The final system – 64 bit Windows*

32 bit users should use Compusec. Once completed the 64 bit user will have

- Unencrypted C drive
- An encrypted pagefile on its own partition
- An encrypted partition containing all the users personal files

2.3 *The final system – openSuSE 10.3*

Once finished the user will have:

- Unencrypted boot partition
- Encrypted root
- Encrypted home
- Encrypted Swap

2.4 *Will my system be fully secure?*

When switched off/ not logged in then yes, however you will still be vulnerable to Viruses, Trojans, Root Kits, internet hackers etc whilst the system is running, and so you should still install all the usual software (virus checker etc) and take all the usual precautions that you would for an unencrypted system.

² Depending on options followed. If not encrypted drive will be wiped of all data at each system shutdown

2.5 How could information leak onto the unencrypted Windows Partition?

If the software you are using was written for Windows XP or later then it is very unlikely that any information will leak out using the system described in this howto.

All user data in newer software saved to the hard disk will go into the users 'Documents and Settings' folder, which will be encrypted. Temporary information will be kept in RAM (and so be wiped at shutdown), or in the encrypted pagefile.

However software written for Dos/ Windows 95/ 98 etc may save temporary data to its own folder on the C drive, so when using legacy software check on temporary file settings, check to see what it saves to disk (maybe it won't be secure data) and see if you can install it on the encrypted drive.

3 Setting up

3.1 Getting started

As it is likely that the system you are reading this on is the one you want to work on it's best to download all the required software now whilst the system is still working. Also print off a paper copy of this howto.

You will need (adapting this list for your needs if you only wish to do a single OS Windows or Linux system):

- A copy of openSuSE 10.3 on CD or DVD (<http://www.opensuse.org>)
- A Windows XP installation CD³ with serial number
- XP drivers for your hardware
- Truecrypt install file for Windows (<http://www.truecrypt.org/>)
- Truecrypt sourcecode for Linux (<http://www.truecrypt.org/>)
- Eraser (<http://www.heidi.ie/eraser/>)
- SwapFS Driver (<http://branten.se/nt/>)
- WinRAR (for unpacking TCGINA only - <http://www.rarlab.com/>)
- TCGINA (for 32 bit system only - <http://www.truecrypt.org/third-party-projects/tcgina/>)
- Ext2IFS (<http://www.fs-driver.org/>) – XP 32bit only
- Ext2fs (<http://ext2fs.sourceforge.net/>) – XP 32 or 64 bit (I prefer Ext2IFS for 32 bit)
- Bootpart (only if you intend to use XP bootloader rather than Grub - <http://www.winimage.com/bootpart.htm>)
- CryptoSwap Guerrilla (optional – <http://www.geocities.com/phosphor2013/list.htm>)

3.1.1 Checking the downloads

If you've had to make your own Windows XP install disk its best that you check it before formatting your current system. All I can suggest though is that you try it out on a spare HDD, computer or virtual machine (e.g. VMWare).

The openSuSE 10.3 CD and DVD images come with a media checking utility. Boot the system from the DVD or CD number 1 and select install from the first menu. Once the system is up (there will be a few more questions) there is a screen that you can use to check each CD/ DVD for errors (which often occur during download). If there are any don't install until you have a good copy.

For all the other software extract and install it on your current system (if you can, - you may not have an ext2/ 3 or swap partition to check swapfs or Ext2IFS on - other wise just check that the files extract ok and install what you can), if any errors occur download fresh copies.

³ If your computer didn't come with an XP install CD, or has a recovery disk which you know will wipe the whole hard disk, then a guide to making a new disk can be found here: <http://www.howtohaven.com/system/createwindowssetupdisk.shtml>

Burn all the software and any personal files you want to keep to CD (or place on any other generic medium).

3.1.2 Determine C – drive required size

Take a look at how much space has been used on your Windows C drive minus the documents and settings folder (which will go on a separate partition), this will help determine the appropriate size for the new C partition.

Make sure you have saved everything you want to keep; once the next step has been completed it will be very difficult to ever recover your data.

3.2 Wiping all data from the HDD

As everyone knows data on a hard drive isn't deleted until it is overwritten.

Even if you have an encrypted file system now it may be possible to look past that at what used to be on the drive, so if there is anything currently on your system you want secure its time to overwrite the whole hard disk.

This is the first step in erasing the old system, as you go through this howto you will gradually ensure that all data you currently hold unsecured on your computer is removed.

There are commercial and free packages to do this (Eraser for one) and I would recommend following the Eraser instructions to make a Boot Nuke Disk.

Another faster way to wipe the whole disk is using the openSuSE rescue system.

Boot openSuSE and from the first set of options select 'Rescue System'.

Once the system has loaded you will have to log-in. The username is 'root' (as in the superuser account), there is no password.

Once in you will have to determine what the name of the HDD is. Normally it is called /dev/hda (IDE) or /dev/sda (SATA/ SCSI), if you don't know make sure all USB etc drives are unplugged (to avoid confusing outputs) and run the following command:

```
Rescue:~# fdisk -l
```

A printout of the result will look something like this:

```
Disk /dev/sda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x31c492b4
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	9729	80026361	7	HPFS/NTFS

In the above example the name of the main hard disk is /dev/sda (/dev/sda1 is a Windows partition on the disk). You are now going to overwrite the entire hard disk with zeros. This form of data deletion won't beat the CIA but will defeat most data thieves, and besides you are going to be doing more to erase your data as you go through. Make sure you have performed a full backup before executing the next command:

```
Rescue:~# dd if=/dev/zero of=/dev/sda bs=4096
```

The option bs=4096 sets the block size for the overwrite. By setting this to an optimal size for your system you can speed up the erase, but if you don't

<http://www.shappyhopper.co.uk/b2154/sharedencryptedhowto.cgi>

Ed Hopper – October 2007

The secured dual boot laptop/ desktop howto.

know what you are doing then it's best to leave it at 4096, this size works for most modern systems.

The erase will probably take an hour or more for an 80 GB drive, and will destroy not only all data but the partition table too.

If you intend to only install Linux go to section 5 on page 16.

4 Windows – Part 1

4.1 Installing Windows

You must install Windows before Linux; most XP installation disks will hang if there is a Linux partition table on the HDD.

The installation of Windows is the same as the default installation, but when you get to the section selecting partitions should you create a new one (NTFS – full format not the quick one), and make it the minimum necessary size you need (this is a necessary step even if you are installing a single OS Windows system).

As there will be no pagefile on the partition you can make the new partition two or three GB larger than the size you determined in section 3.1.2.

As an example; an XP 32 bit installation with office 2003 (full plus Visio) and other standard packages like Acrobat Pro, requires a 10GB partition, of which 3.5GB will left over after installation of all packages.

4.1.1 User accounts

If you have to (for example XP home) only install one user account (XP Pro users should just stick with the default Administrator account), remembering that this account will never be used so do not use your favourite account name.

At this time this account should **NOT** be password protected until we have disabled LM hashing (unless you fancy setting a 2nd password later on) and a computer administrator. If you do set a password it should not be the password you intend to use to unlock your encrypted partitions⁴.

⁴ Note: when using XP home a hidden Administrator account with no password is created. This is disabled by default, if you wish to add a password (after disabling LM hashing) run 'control userpasswords2' and change the password to something else.

4.1.2 Disable LM Hashing

Once the system has been installed log in and disable LM Hashing to secure your XP user passwords. For an explanation of what this is go to one of the following sites:

- <http://tsudohnimh.com/blog/2006/09/secure-passwords-part-two.html>
- http://articles.techrepublic.com.com/5100-6350_11-5287636.html

Click Start, click Run, type regedit, and then click OK. Locate and then click the following key in the registry:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

On the Edit menu, point to New, and then click DWORD Value. Type 'NoLMHash', and then press ENTER (note this DWORD may already exist). On the Edit menu, click Modify, set the value to 1 and then click OK.

Next find 'LMCompatibilityLevel' and set to one of the following – depending on your system (if you don't need to log into a Windows network server just set it to 5):

- Level 0: Send LM response and NTLM response; never use NTLMv2 session security
- Level 1: Use NTLMv2 session security if negotiated
- Level 2: Send NTLM authentication only
- Level 3: Send NTLMv2 authentication only
- Level 4: Refuse LM authentication
- Level 5: Refuse LM and NTLM authentication; accept only NTLMv2

Restart your computer and then change/ set your passwords.

4.1.3 Get system up and running

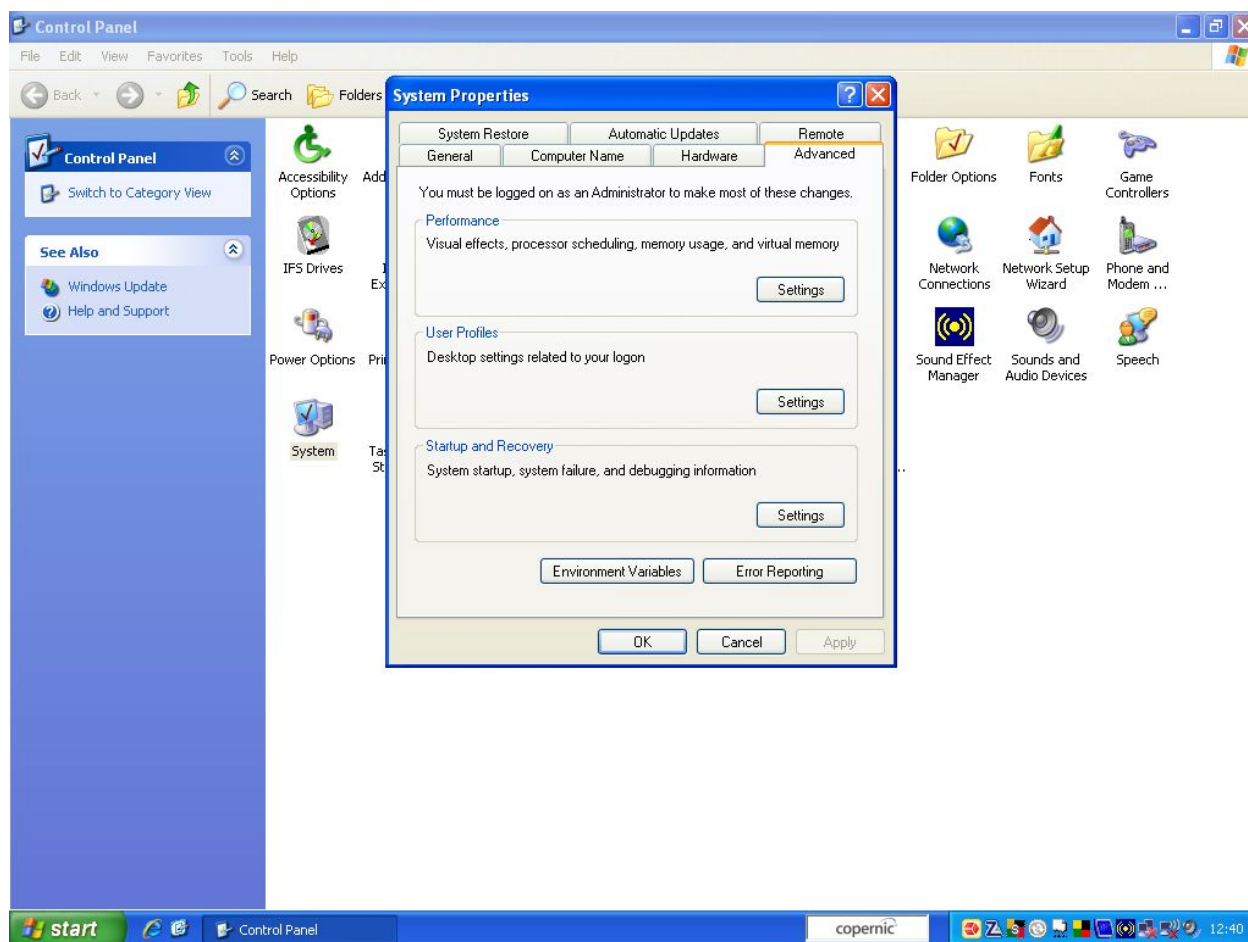
Now install all your Windows drivers to get the system up and running. If you do go online or open any files make sure none of them are sensitive as the account you have made will never be secure and you don't want to risk fragments of sensitive data ending up on this drive⁵.

4.1.4 Check the pagefile size

Once you have installed Windows you need to know the size of the pagefile it is using, this will help to determine the size of the openSuSE swap partition, that both Windows and Linux will use.

To find the page file size go to 'Control Panel => System' and select the advanced tab.

⁵ The free space on this drive will however be wiped later in this howto.



Click on the settings button at the top, the one inside the 'Performance' box. Now select the advanced tab on the 'Performance Options' window, and then select the 'Change' button within the 'Virtual Memory' box at the base of the window.

At the top of the 'Virtual Memory' window you will see a white box in which is listed the C: drive and its Paging File Size. Make a note of the maximum size.

4.2 Install Software

Finally install Truecrypt, WinRAR, Ext2IFS/ Ext2fs and Eraser, you will need them later.

If you intend to use bootpart install that now too, however please note that using the Windows XP boot loader can cause issues with the advanced encryption techniques described in section 7.4 (for an explanation see section 7.4.3 on page 34).

4.3 Single OS Windows System

If you are not going to install Linux you should now also make a 2nd and 3rd partition. The first partition will be for the pagefile, and should be a similar size to the Linux swap partition described in section 5.1.1 on page 16.

The final partition should take up the rest of the drive. This should be formatted as NTFS with Truecrypt, it will become the documents and settings folder.

You can now skip ahead to section 6 on page 27

5 Linux – Part 1

5.1 Installing openSuSE 10.3/ Linux

Boot the machine with the openSuSE DVD/ CDROM. Follow the default install until you get to the system configuration screen. This is the screen which will show you a summary of the system to be installed, such as software and partitioning.

5.1.1 Partitioning

First you change the partitioning. As the HDD has a large chunk of free space (or is just free space if you are only installing Linux) openSuSE will have determined optimal partition sizes, sadly you want none of these but before you delete them make a note of the swap partition size. Compare this to the Windows page file and whichever is the larger will become the new swap partition size +30ish % (so if your pagefile needs to be at most 1.5GB the swap partition will be 2GB).

If there is no Windows system on the disk just note and use the default swap size.

Once all the default partitions have been deleted you must first create a boot partition (so set the mount point to '/boot'). This should be 75MB as you may want to have more than one kernel and initrd.

Next you create the swap partition, size as above, followed by what will eventually become the root partition. My partition is 8GB and Ext3, set the mount point to nothing.

The remaining space on the disk will be for documents. For now you will install Linux there (so set the mount point to '/') but you will move that to the new root partition later.

Install the root file system ('/') on one of them and leave the mount point of the other one blank. It doesn't matter at this time what file system you format this as it will be overwritten by Truecrypt, however whatever it is will be the final file system of the installation (as we will create a file system within the encrypted devices).

On our example system my partition table now looks like this:

Device	Size	Type	Filesystem	What is it?
/dev/sda1	10GB	Primary	NTFS	Windows C
/dev/sda2	70GB	Extended	W95 Ext'd (LBA)	Extended
/dev/sda5	75MB	Extended	Ext3	/boot
/dev/sda6	2GB	Extended	swap	swap
/dev/sda7	10GB	Extended	Ext3	Future root
/dev/sda8	58GB	Extended	Ext3	Root ('/')

5.1.2 Grub

If you intend to use hibernate you should use the openSuSE default boot manager grub (the reason is discussed in section 7.4.3 on page 34), and unless you have a reason to use the XP boot loader you should leave the bootloader options alone.

However if you have security software that requires you to have an XP MBR you should set the installation location of GRUB to your /boot partition (in this case /dev/sda5) use bootpart to add openSuSE to your Windows XP bootloader (and also see section 7.4.3 on page 34).

If you have no idea what all this means leave Grub/ the bootloader alone for now.

5.1.3 Software

You should now add some additional packages during installation. Select the dropdown to change software, in the screen which appears click on 'Details' (near the bottom of the screen) and on the next screen which appears select 'Search' from the dropdown near the top.

Now search for and select the following packages:

Package	Why
kernel-source	Needed to build truecrypt module
kernel-syms	Needed to build truecrypt module
cryptconfig	Needed for encrypted root FS
cryptsetup	Needed for encrypted root FS
libgcrypt	Needed for encrypted root FS
libxcrypt	Needed for encrypted root FS
gcc	Needed to build truecrypt module
make	Needed to build truecrypt module
e2fsprogs	Needed for encrypted root FS
e2fsprogs-devel	Needed for encrypted root FS
kinternet	Added for convenience, otherwise it will ask to install later

Accept any packages added in support of the above and any licence agreements and return to the summary screen.

It is now ok to run the installation.

5.1.4 Update

Do not run update during installation as a new kernel is already out and this will make the install fail. Wait until the system is running (as in section 5.2).

5.1.5 Root password and user accounts

Besides doing anything else during installation you need to make sure that when you set the root password (and you should) that you don't use the password you will want to use for the encrypted filesystems.

At this time do not add any user accounts, and ignore the warning that this will only be an empty login server (or whatever that warning said).

Once the installation has finished log in as the root user.

5.2 Randomising the root partition

Once you are logged in as the root user on your new system you should start the process of moving root to its new encrypted partition.

However first run YaST, get the system hooked up to the internet, add an Online Update source and run an online update. This will install a new kernel and some security updates (in fact the longer after openSuSE 10.3 came out you read this the more updates there will be).

The newer kernels for openSuSE 10.3 implement (or seem to implement) the use of mkinitrd in a slightly different way to the default installation, and doing this update skips a chapter I would've had to otherwise write.

Once the update is complete reboot and log in again as root.

The first thing you are going to do is overwrite the soon to be root partition with random data. This will help to erase anything that you once had on the disk that wasn't removed earlier, and help hide the encrypted data that the root partition will soon contain.

As can be seen in section 5.1.1 on the example system the new root partition will be /dev/sda7 so you run the command⁶ (from a terminal or terminal emulator):

```
tin-man:~# dd if=/dev/urandom of=/dev/sda7 bs=4096
```

This will overwrite the whole of the new root with data from urandom, a device in the Linux system which generates random data by combining random number generators with information on disk timings and mouse movements. It runs pretty slowly and for a 10GB drive will take several hours to finish. By making the block size (bs=4096) larger you haven't made the process run

⁶ Please note: for this example my Linux machine is named 'tin-man'.

<http://www.shappyhopper.co.uk/b2154/sharedencryptedhowto.cgi>

Ed Hopper – October 2007

The secured dual boot laptop/ desktop howto.

faster (unlike with /dev/zero earlier), but you have reduced the overhead on the system as it writes to disk less often, and so you will be able to do other things whilst this runs.

If you want you can move ahead whilst this runs and do the next step, but as that (the next step) will only take a few minutes maybe you should just get a cup of tea.

5.3 Setting up the bootloader - Grub

Whilst the system is busy writing random data to the new root partition you can be setting up Grub (or enjoying that tea).

Using the file manager as root go to /boot/grub; inside the directory you will find a file called 'menu.lst'.

Open this with your favourite text editor and add a new entry (modified for your system) as shown in the example in appendix B on page 39.

Save the file and close the text editor.

5.4 Encrypting and formatting root

One the random data has been written to the new root you need to create an encrypted file system, run the following command:

```
tin-man:~# cryptsetup -v --key-size 256 luksFormat /dev/sda7
```

And follow the instructions. Remember to use a different password from your usual user account login.

Next you need to open the new partition, run:

```
tin-man:~# cryptsetup luksOpen /dev/hda3 root
```

And enter your password.

Finally you need to format the partition, I use Ext3 here but you should use whatever the original root system was installed on (the default openSuSE 10.3 filesystem is Ext3).

```
tin-man:~# /sbin/mkfs.ext3 -j /dev/mapper/root
```

5.4.1 Making a crypttab file

When the system boots the encrypted file system software looks for a file listing encrypted systems and what they should be booted as. This file is called crypttab and is found in /etc, e.g:

```
/etc/crypttab
```

See appendix C on page 40 for an example crypttab entry, you will need to edit this file for your system and place it in /etc.

5.5 Moving the root Filesystem

Once the new root partition has been created you need to copy over the current root file system.

First mount the partition.

```
tin-man:~# mount /dev/mapper/root /mnt
```

If the directory /mnt does not exist make one using file manager.

Within /mnt make the directory's 'media', 'proc', 'boot' and 'sys'. These are system directories that you don't need to copy over; they are set up as the system is running.

Now you copy the rest of the root partition:

```
tin-man:~# cd /
tin-man:/ find bin dev etc home lib* opt root sbin \
>srv tmp usr var -depth -print0 | cpio -pmd --null /mnt
```

The above command (the 2nd line onwards) can be written as one line, by entering \ at the end of the first line I was just signifying that the command would continue onto another line. The > is inserted at the start of the next line by the system.

The complete command without all this can be found in appendix D on page 40.

5.6 Mount options: fstab

The fstab, which is the file the running system uses to determine where and if to mount partitions, needs to be altered for the new encrypted root system.

Go to /mnt/etc (the /etc within the new root filesystem) and open the file 'fstab' with your favourite text editor.

Appendix E on page 41 shows an example fstab, for the new system you only need to alter one line, the line for the root filesystem.

5.7 The initial Ram Disk - Initrd

The Linux system uses an initial ram disk (initrd) to boot the system, containing and modules and software that the kernel will need prior to mounting the root directory. In order to mount an encrypted root filesystem a few tools must be added to the initrd.

With the introduction of openSuSE 10.3 the system will run almost out of the box with the standard kernel and software

Run the following command:

```
tin-man:~# mkinitrd -d /dev/mapper/root
```

Which tells mkinitrd to make a new initrd based on the root system on /dev/mapper/root.

Once done you should be able to reboot the system.

5.8 The first boot, what to do if it doesn't work, and what to do when it does

When you reboot you should see the option to boot "Encrypted openSUSE 10.3", if not you must go back in and edit the grub menu as per section 5.3 on page 20.

Next you should see the usual openSuSE booting screen (with the option to press Escape for more information). After a few seconds (which will seem longer because you are waiting and watching) the screen will show the boot process, and the words "Enter Luks password:" should appear.

They may have appeared and be a few lines back, because on some systems the kernel will continue to discover hardware and report on it. Either way just enter the password you made up in section 5.4 (page 20) – no stars or text will appear - and press return, the system should then continue to boot.

If you are unable to type anything re-boot Linux via the normal way and check that your keyboard is supported by initrd during boot. USB keyboards commonly are not; you will therefore need to find a guide on the web to enable the keyboard. Commonly you can enable USB keyboard support in your initrd with the command (all as one line):

```
tin-man:~# mkinitrd --with=ehci-hcd --with=uhci-hcd -d /dev/mapper/root
```

Or add the modules to initrd using YaST and remake the initrd for /dev/mapper/root.

If you get error messages like 'no filesystem driver' or similar then initrd needs to be rebuilt; log back in and run mkinitrd as per section 5.7 and look for error messages, if it runs without incident try rebooting again.

Sometimes initrd needs a bit of a kick to make it work, in appendix A on page 38 there is a dummy script that you can use to try to give it a push. I based it on the work of someone else, it actually does nothing useful but may help mkinitrd to realise it has to add a script and therefore must completely rebuilt initrd.

Once you have booted you should run the command 'mkinitrd' (and just that word) to make sure that the new system is able to build its own initial ram disk (this will be needed every time a kernel update is installed).

5.9 Creating the /home directory

If your system has booted up successfully then you can proceed to create a '/home' directory.

If you want to have the '/home' partition visible to both Windows and Linux as a shared documents and user settings partition you will need to install Truecrypt.

5.9.1 Installing truecrypt

Unpack the contents of the truecrypt source code tar you downloaded (section 3.1) into your home directory (which for root will be /root). Make sure that the full file path to where you unpack the source code has no spaces in the name; otherwise the kernel module build will fail.

From a terminal or terminal emulator enter the 'Linux' directory within the unpacked source code directory and run:

```
tin-man:~# ./build.sh
```

Followed by:

```
tin-man:~# ./install.sh
```

In theory just running ./install.sh should work, but I have found that if there is already an existing built module in the directory the install script will just try and load that, which may be problematic as it will give no errors but could affect other running kernel modules.

After you have installed Truecrypt check module dependencies (also run this before making a new initrd (command mkinitrd) to prevent errors regarding the module 'dm-truecrypt').

```
tin-man:~# depmod -a
```

5.9.2 A note about Truecrypt and new kernel updates

You should keep the source files in your root home directory as you will need to rebuild Truecrypt (using both commands in section 5.9.1) after every kernel update. Because you have installed kernel-symbols the module should always load and allow you to mount /home, even with the different kernel; however you may have odd hardware errors (on mine WIFI drops out) due to module conflicts so you should always re-build Truecrypt (it takes seconds after all) and run 'depmod -a'.

Keeping it in the root partition ensures that it is always available, even if Truecrypt won't mount /home anymore.

5.9.3 Remove old items from grub

The blue highlighted text in the example in appendix B on page 39 (or whatever is similar for your system) can now be deleted from the grub menu. You may want to reboot to check its all working.

5.9.4 Creating the home directory

Now that the root system is installed and working it's time to delete the old root file system, which in your example is on /dev/sda8.

Run the command:

```
tin-man:~# truecrypt -c /dev/sda8
```

You will be asked what volume type you want, select 'normal', and when asked which filesystem you want, 'fat' or 'none' select 'none', follow the rest of the instructions to create an encrypted file system.

There is no need to randomise this partition first as truecrypt will fill it with random data, this will take an hour or more depending on system speed and partition size. Again go and make some tea.

5.9.5 Formatting the home directory

Mount the new partition with the command:

```
tin-man:~# truecrypt /dev/sda8
```

This should mount the partition as /dev/mapper/truecrypt0, to format run the command:

```
tin-man:~# /sbin/mkfs.ext3 -j /dev/mapper/truecrypt0
```

If you don't intend to share this with windows then you can format the partition with any Linux filesystem (e.g. reiserfs).

Next un-mount the volume:

```
tin-man:~# truecrypt -d /dev/sda8
```

And run a test install replacing the word 'yourpassword' with the password you used to create the truecrypt volume:

```
tin-man:~# truecrypt /dev/sda8 /home -p yourpassword
```

5.9.6 Setting up an auto-mount for /home

You now want to set the system up to mount this new truecrypt partition at boot, to do this edit the file /etc/init.d/boot.local as shown in appendix F on page 42. Reboot the system and check that the '/home' drive is mounted (the '/home' folder will go from being empty to containing a folder called

'lost+found') Also keep an eye out to make sure that disk checking is run every so often..

5.10 Adding a user account

Now that the system has been created with an encrypted '/' (root) and '/home' you can add a user account. Do this using YaST.

If you have enabled the auto-mount option for the Truecrypt partition, and have your password in the boot.local file you can now set up the user account to log-in automatically. This is safe as the entire basic Linux system is secure (except for swap which is covered soon), and it means that you will only need to enter one password between boot and desktop!

That's the basic Linux system set-up, with the exception of the swap space, which you will deal with later.

If you are only installing Linux you can move on to section 7, page 29.

6 Windows - Part 2

6.1 *Installing TCGINA and encrypting the Windows user account*

Now that Truecrypt and WinRAR have been installed (section 4.2 on page 14) you can extract and install TCGINA. Once installed go to the start menu and run:

“control userpasswords2”

Click on the advanced tab and then check the box “Require users to press Ctrl+Alt+Delete”, then click Ok and reboot the machine. This option is set to support TCGINA.

Once you have rebooted create a new user account – the one you actually want to use, log off, log into the new account, log out and log back into the first account.

Make sure the password you use for the new account isn't the same as the one you use for either the Truecrypt or Linux root partitions.

Run Truecrypt. If you are sharing a partition with Linux mount the shared encrypted partition on a high drive number, I use Z. This is because this drive must not change if you are to log into it as a user, and if you map the Truecrypt drive to drive E, then boot with a USB stick in the computer you will be unable to log-in.

Now run the TCGINA install program again and you should be able to move your user account to the encrypted partition. Next time you log into your new account you will log-in first via the Windows login, and then you will be asked for the Truecrypt partition password.

Once all this is done reboot the computer and log into Linux, open the root user account and go to '/home'. A new folder will be there called 'Documents and Settings' change the permissions and group to match your Linux user account and set it to do this to all sub-folders and files.

Once done you can log back into Windows and move 'My Documents' to a convenient folder within your Linux home directory, I just use a folder called 'Documents' which is installed by default in the openSuSE system. Whenever you create a file in Windows now it will be saved with the permissions of the user account in Linux.

6.2 Moving the Windows pagefile to the Linux Swap partition

The SwapFS driver package can now be installed. Follow the instructions for installing the driver, when it comes to editing the registry entry Truecrypt can help.

If you are not using a dual Linux Windows system just follow the instructions below to move the pagefile to the partition you created for it.

Whilst on the Linux system the swap partition is partition 6 (/dev/sda6) under Windows it is '\\Device\\Harddisk0\\Partition3'. If you run Truecrypt and click on 'Select Device' you will see a list of partitions as Windows sees them, and from that you should be able to determine the swap partition as Windows sees it and edit the registry entry correctly.

You can change the name of the drive in the last line from 'S:' to whatever you want, I use 'Y:' so it sits next to the Truecrypt partition.

An example can be seen in appendix I on page 45.

Re-boot the computer, if the registry entry was correct you should now have a new drive with a Fat file system. Go to the pagefile settings window as per section 4.1.4 (page 13), select drive Y: (or whatever you set yours as) and set the pagefile size, then un-set the page file from the 'C:' drive. If the drive appears with a '?' next to it in explorer check your settings and edit the registry.

As an additional security option you should set the minimum and maximum sizes of the pagefile to the original maximum. This ensures that each new page file completely overwrites the last, rather than a growing/ shrinking file receding and leaving confidential data on the swap drive for months. More security options for the pagefile are discussed in section 7 on page 29, including how to delete or encrypt it.

6.3 Wiping the free space on Windows

You now need to ensure that any old data on the Windows C: drive (before you started this whole process) is erased.

Open power properties from the control panel and disable hibernation (which will delete hiberfil.sys). Now run defragment on the C: drive; as can be seen now that the pagefile and hibernation files are gone there is no longer any unmoveable data on the drive.

Next run Eraser. You might want to reboot after the defragment as sometimes Eraser can cause errors when run on a recently defragmented drive. Set eraser to erase all free space on the C: drive and run, it will take 30 mins to several hours to run, depending on system and disk size.

7 Encrypting the system swap and hibernation space

The final part of this howto covers the system swap and hibernation spaces. There are several options here for system security, starting with the most basic and finishing with high level security options.

The higher security options, whilst providing a significant level of security, can carry with them a much higher maintenance requirement than the simple options.

7.1 Swap Space

On Windows this is the pagefile, and under Linux this is the swap partition.

Swap space is used when the computer runs out of physical memory (RAM), to free up physical memory information is swapped out of RAM into the swap space.

Therefore, even though you have encrypted all of your personal files, folders and personal settings (and the Linux root) there is still the risk that sensitive data will leak out into the swap space.

Of course if you are confident that your system has so much RAM it will never use swap space, and you will never want to suspend Linux to disk then you can set Windows to have no pagefile, and delete the Linux swap; for the rest of you, you have to look at either deleting or encrypting this data.

7.2 Hibernation

When a system hibernates it saves the contents of RAM into a file on the hard disk. In Windows this file is always on the C: drive (it cannot be moved) and is called hiberfil.sys. On Linux this is a RAM disk image stored on the swap partition.

As above there is a risk that sensitive data will be saved to the hard disk, in fact there is a greater risk as you will be deliberately dumping the entire contents of RAM to the disk.

On very secure systems its best to disable hibernation altogether on both systems, if you want this feature you have to look at encrypting this data too.

7.3 Simple steps – Wiping/ encrypting Swap and deleting/ encrypting the pagefile

Your system is already more secure than a standard system; by having both Windows and Linux write to the same swap space (see section 6.2 on page 28) you are ensuring that anything on the partition is regularly overwritten with data of different formats.

Another basic step you can do is have the Linux system wipe the swap partition on reboot, and have the Windows system delete the pagefile.

Skip to section 7.3.4 if you intend to use the advanced techniques discussed later.

7.3.1 Linux – wipe the swap partition

This will add up to a minute to the shutdown time (with a 2GB swap, and about 30s with a 800MB swap) but will provide an extra level of security when combined with the delete pagefile below.

You are going to add a series of commands to the `/etc/init.d/halt.local` script which will be executed as root once the system is almost shutdown.

Open the file with your favourite text editor and add the following lines:

```
swapoff /dev/sda6
dd if=/dev/zero of=/dev/sda6 bs=4096
mkswap /dev/sda6
swapon /dev/sda6
```

You may want to try executing the 'dd' command from a terminal as root (after first making sure you have switched off swap) using different block sizes (the 'bs' option) to see what is the optimal setting for your system, and hence speed up the shutdown process. However even with the optimal setting you will still see a significant increase in the shutdown time.

7.3.2 Windows - Delete the pagefile on reboot

This is achieved by a simple registry change; it isn't necessary if you use the feature above to wipe the swap partition and regularly boot Linux.

Go to the start menu and run 'regedit' then navigate to the key:

```
H_KEY_LOCAL_MACHINE\SYSTEM\CURRENTCONTROLSET\CONTROL\
SESSIONMANAGER\MEMORYMANAGEMENT
```

Set the key 'ClearPageFileAtShutdown' to 1. This will add an extra delay to the Windows shutdown whilst the pagefile is deleted.

7.3.3 Linux – simple encrypted swap without hibernate or Windows pagefile share

I've included this for completion, but it goes very much against the idea of this howto. If you need to have an encrypted swap there is an example later of how to have an encrypted swap with pagefile partition sharing and encrypted hibernate.

First of all log into Windows and move the pagefile to another partition. Then log into Linux and run YaST. Open the partitioning tool and format the swap as an encrypted partition. You don't have to enter a password, and if you don't a new encryption key will be created each time you boot based on /dev/urandom.

That's it, an encrypted swap which even the system itself will be unable to access after reboot, which is why hibernate will no longer be available, and even if you enter a set password the Kernel will not be able to load the partition at boot, hence, again, no hibernate feature.

7.3.4 Windows – Encrypting the pagefile on the fly

For the person who wants an easy life (before you get on to the more advanced security stuff) is the free software package CryptoSwap Guerrilla which will encrypt the Windows pagefile on the fly. It is recommended that you still use this package with a separate partition (such as the shared swap) rather than keeping your pagefile on the C: drive.

If you want to use the expert option (section 7.4) you should use the shared swap partition and CryptoSwap Guerrilla.

Note: by default CryptoSwap will enable the Windows option to delete the pagefile on shutdown/ reboot (section 7.3.2). This is not really necessary as the pagefile data is encrypted, and Linux will format the partition on boot anyway; it will just make shutting down take longer.

Luckily CryptoSwap contains a standard registry entry to disable this, which can be found in the directory you installed CryptoSwap to (e.g. 'C:\Program Files\CryptoSwap Guerilla').

7.3.5 Linux – Encrypt the hibernate disk image

As part of the standard openSuSE 10.3 installation is the option to encrypt the hibernate RAM disk image. The image is still stored on the swap partition, and the partition is still unencrypted, however the image itself can't be read.

To enable this open `/etc/suspend.conf` with a text editor; scroll down and un-comment (remove the `#` at the start of the line) the following options:

- `Encrypt=Y`
- `RSA key file=/etc/suspend.key` (or whatever the keyfile is to be called)

An example `suspend.conf` file can be seen in appendix J on page 46.

When you now suspend to disk the system will ask you for a password, and then write it to the file specified in the RSA Key file option. As the root file system is encrypted this is very secure.

If you want to use the expert option (section 7.4) you should enable this feature.

7.4 Expert steps – Encrypted pagefile, swap and Linux Hibernate

This section is for experts or the very paranoid who know how to carry out maintenance on their system if an error occurs during a system change, such as a new Kernel via online update.

There is no need to read this section if you are only using Windows.

It's not that this method causes any errors under the current openSuSE 10.3 system, but you will be regularly changing the format of a key system partition and this may cause a boot failure following future system updates. You should be comfortable dealing with such issues before enabling these options.

7.4.1 Warning

Please note; if your system regularly uses large chunks of the Linux swap partition (for example if it is low on resources), or is likely to just prior to the hibernate process then do not use this method as errors will occur when resuming and work can be lost. This method is particularly not recommended for systems with limited resources.

A swap space monitoring package (several are provided with openSuSE 10.3) will tell you what your usage is like.

Most systems however should have no problem.

7.4.2 A note on Windows hibernate

At the time of writing there were no free software packages that could encrypt the Windows hibernate file.

The hiberfil.sys file also cannot be moved from the C: drive so unless you purchase software to encrypt the C: drive there is no way to securely operate Windows hibernate, and if your system needs to be fully secure disable the feature (Control Panel=> Power=> Hibernate Tab).

If you do need to use hibernate then I recommend that you regularly disable hibernate (which will delete hiberfil.sys) defragment and run Eraser as described in section 6.3 on page 28 and then re-enable hibernate.

Alternately install a 2nd physical hard disk for Linux (if an option) and use CompuSec to encrypt the whole Windows physical drive.

7.4.3 Boot manager – use Grub

I recommend that you use the Grub boot manager and not the Windows boot manager when using the encrypted hibernate option.

This is because if you hibernate your Linux system and then boot Windows the Windows pagefile will overwrite the RAM disk image created by Linux, and you will lose whatever you were working on in Linux. If you use the Grub boot manager you are forced to resume and shut down Linux before booting Windows.

Alternately just remember not to boot Windows, or move your pagefile to an alternate partition.

7.4.4 Creating an encrypted swap system, compatible with pagefile sharing and using encrypted Linux hibernate

Before you start install CryptoSwap Guerrilla and configure Windows pagefile as in section 7.3.4 on page 31. Also enable the encrypted hibernate option in Linux as per section 7.3.5.

This method will change the state of the swap partition when booting, hibernating and shutting down.

When the system is running the swap partition will be encrypted with a random keyfile created from `/dev/urandom`.

When it is switched off the swap partition will be clean formatted and unencrypted; the only data that can be recovered from it will be fragments of encrypted data from the Windows and Linux systems.

When the system is hibernated (Linux only) it will be a freshly formatted swap partition with an encrypted RAM disk image on it.

Sadly because the swap/ resume partition presented to the Kernel at boot, and whilst running, is fundamentally different you may have to do some re-configuring of your system during certain updates. Ensure you are comfortable doing this before enabling this option.

7.4.4.1 Getting started

Configure the system to use CryptoSwap Guerrilla and the encrypted Linux hibernate image (sections 7.3.4 and 7.3.5 on page 31).

The following steps must be followed before rebooting the system.

7.4.4.2 Remove swap devices from /etc/fstab

Open fstab in a text editor as root and delete the line referring to the swap partition; see the example in appendix E on page 41.

7.4.4.3 Edit /etc/init.d/boot.local

You are going to edit the local boot script (which contains your Truecrypt mount command) to create and mount an encrypted swap partition.

An example boot.local can be found in appendix F on page 42.

7.4.4.4 Edit /etc/init.d/halt.local

Next edit the local halt script to turn off swap, dismount the encrypted swap partition and format it as a standard swap partition.

An example halt.local can be found in appendix G on page 43.

7.4.4.5 Create a script for hibernating the computer

User scripts used during the hibernation of a computer are stored in the directory:

- /etc/pm

These are executed along with the system scripts in numerical order, based on the name of the script (which can be between 00 and 99, however most have been used). If you do not know what you are doing with power management just follow the exact instructions here (noting that they are for a default openSuSE 10.3 system).

To create the script enter the directory:

- /etc/pm/sleep.d

And create a file called '04swapencryption'. An example showing the contents of this file can be found in appendix H on page 44. Make sure you edit the script to match your swap file partition.

7.4.4.6 Filling the swap partition with random data

You have now ensured that no more un-encrypted data will ever be written to the swap partition so you should now ensure that any old data from the previous system on this computer is overwritten.

Run the following commands and then reboot, once up the system should now be fully encrypted, with the exception of the Windows C: drive.

```
tin-man:~# swapoff /dev/sda6  
tin-man:~# dd if=/dev/urandom of=/dev/sda6 bs=4096
```

As before this will take some time, but setting the block size to one suited to your system should make it possible to continue working until the command is finished. Of course with no swap the system may be slow.

8 Bugs and issues to note

Now that you have done all that work I should mention a few of the bugs you may experience.

8.1 Unable to mount Truecrypt in Windows after Linux crash and vice-versa

When using a shared encrypted /home drive:

If the system dies unexpectedly (i.e. unplugged or switched off) you will have to re-boot into the operating system you were using prior to the crash otherwise Truecrypt won't mount the shared drive until you re-mount it from that OS. Minor crashes can sometimes happen during shutdown, if they happen regularly add a command to halt.local to manually unmount the Truecrypt partition before the system is halted (an example is given in appendix G on page 43).

8.2 Resume fails or data is missing after hibernate

This can happen when using the system described in section 7.4 it happens because a large amount of data was still on the swap partition prior to the hibernate script wiping the partition. If this happens regularly you should either stop using this encryption technique, stop using hibernate, reduce the system overhead before hibernating (e.g. stop any unnecessary processes or do not use any particularly onerous packages) or install more RAM.

8.3 Spyware software doesn't work

At least one free spyware package, Spyware Doctor (free via Google Pack) will have errors when installing and loading on this system.

However AVG Anti-Spyware (free) and Windows Defender work fine with this system.

A. Dummy initrd script

When I was trying to work out how to get an encrypted root to work I came across this - <http://osdir.com/ml/swsusp.general/2005-03/msg00017.html> - page where someone had posted a script to alter initrd (below). I placed it into the directory `/lib/mkinitrd/scripts` and called it `'zz01_root_encrypted.sh'` to make sure it was never overwritten by any system updates. Low and behold I found that `mkinitrd` had worked. I later realised this was because I had been running `mkinitrd` incorrectly; however you can give it a go if you find your `initrd` not booting.

```
#!/bin/bash

# Mot de passe unique pour tous les montages dans initrd

cat <<EOF >${INITRDDIR}/keyscripts/one-pass.sh
#!/bin/sh

if [ ! -f /dev3/cryptpass ]; then
    mount -n -t tmpfs tmpfs /dev3
    echo
    echo " ~~~~~~ "
    echo " This system uses encrypted disk "
    echo " ~~~~~~ "
    echo -n "Mot de passe : "
    stty -echo
    read cryptpass
    echo \${cryptpass} > /dev3/cryptpass
    stty echo
    echo
fi

# You decrypt the disk
echo "Decrypting \${dmname}..."
cat /dev3/cryptpass | /sbin/cryptsetup -v -c \${cipher} mode create
\${dmname}
\${device}

# If it is swap, you try to resume
if [ "\${dmname}" = "swap" ]; then
    /resume.sh
fi
EOF

chmod +x ${INITRDDIR}/keyscripts/one-pass.sh
cp /bin/stty ${INITRDDIR}/bin/stty

# Later, you need to umount /dev3

cat <<EOF >${INITRDDIR}/scripts/umount-dev3.sh
#!/bin/sh
umount -n /dev3 2> /dev/null
EOF

chmod +x ${INITRDDIR}/scripts/umount-dev3.sh
mkdir ${INITRDDIR}/dev3
```

B. Grub entry

The following green highlighted text is an example grub entry for use whilst testing the new root filesystem. The blue highlighted text will be removed from the grub menu following a successful boot and the green text will become the default boot menu item.

```
# Modified by YaST2. Last modification on Thu Oct 11 09:58:38 UTC 2007
default 0
timeout 8
gfxmenu (hd0,4)/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title openSUSE 10.3
    root (hd0,4)
    kernel /vmlinuz-2.6.22.5-31-default root=/dev/sda8 vga=0x317 resume=/dev/sda6 splash=silent showopts
    initrd /initrd-2.6.22.5-31-default

title Encrypted openSUSE 10.3
    root (hd0,4)
    kernel /vmlinuz-2.6.22.5-31-default root=/dev/mapper/root vga=0x317 resume=/dev/sda6 splash=silent showopts
    initrd /initrd-2.6.22.5-31-default

###Don't change this comment - YaST2 identifier: Original name: Windows###
title Windows
    rootnoverify (hd0,4)
    chainloader (hd0,0)+1

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe -- openSUSE 10.3
    root (hd0,4)
    kernel /vmlinuz-2.6.22.5-31-default root=/dev/mapper/root vga=normal showopts ide=nodma apm=off acpi=off
    noresume nosmp noapic maxcpus=0 edd=off 3
    initrd /initrd-2.6.22.5-31-default
```

C. Cryptab Entry Example

The following is a one line entry in the file `/etc/crypttab`. The file can be made with any text editor, e.g. `kwrite` and should be edited for your system and placed in `/etc`. The first word 'root' refers to the device name it will be given in `/dev/mapper`, hence for this example the device would be `/dev/mapper/root`.

```
root /dev/sda7
```

D. Root FS move command

The command written in section 5.5 on page 21 had to be re-formatted due to the width of the page, the full command is as follows:

```
tin-man:~# find bin dev etc home lib* opt root sbin srv tmp usr var -depth -print0 | cpio -pmd --null /mnt
```

If there are additional directories in root (`/`) then enter them after the word 'find'. Do not include 'boot', 'media', 'sys' or 'proc', these can be made as blank directories and will be set up by the running system.

E. Fstab example

The following is an example of an altered fstab for the initial boot of an encrypted filesystem. The green highlighted part is what the entry for the root filesystem should look like as per this example. The rest of the line can remain unchanged.

The blue highlighted line should be deleted if you are following the instructions in section 7.4.4.2 on page 35.

```
/dev/mapper/root / ext3 acl,user xattr 1 1
/dev/disk/by-id/scsi-SATA_WDC_WD800UE-00H_WD-WXE105258152-part5 /boot ext3 acl,user xattr 1 2
/dev/disk/by-id/scsi-SATA_WDC_WD800UE-00H_WD-WXE105258152-part6 swap swap defaults 0 0
proc /proc proc defaults 0 0
sysfs /sys sysfs noauto 0 0
debugfs /sys/kernel/debug debugfs noauto 0 0
usbfs /proc/bus/usb usbfs noauto 0 0
devpts /dev/pts devpts mode=0620,gid=5 0 0
```

F. /etc/init.d/boot.local example - Truecrypt

The following green highlighted text is an example of what to add to the boot.local file to have your truecrypt partition auto mounted at boot. If you remove the `-p` and `yourpassword` (which should be the volume password) you will be asked during boot to enter your volume password. There is little risk keeping the password in plain text form in boot.local as the root filesystem is encrypted.

The blue highlighted section should be added if you are following the instructions in section 7.4.4.3 on page 35.

```
#!/bin/sh
#
# Copyright (c) 2002 SuSE Linux AG Nuernberg, Germany. All rights reserved.
#
# Author: Werner Fink <werner@suse.de>, 1996
#         Burchard Steinbild, 1996
#
# /etc/init.d/boot.local
#
# script with local commands to be executed from init on system startup
#
# Here you should add things, that should happen directly after booting
# before you're going to the first run level.
#
#You first encrypt the swap partition
cryptsetup -c blowfish -h sha256 -d /dev/urandom create swap /dev/sda6
mkswap /dev/mapper/swap
swapon /dev/mapper/swap

#And now you mount the truecrypt partition, ensuring that a disk check is run
truecrypt /dev/sda8 -p yourpassword
e2fsck -p -v -C0 /dev/mapper/truecrypt0
mount /dev/mapper/truecrypt0 /home
```

G. Example /etc/init.d/halt.local

The following example halt.local is to be used if following the instructions found in section 7.4.4.4 on page 35. The highlighted section is to be added.

The red highlighted entry is optional, and used to manually unmount the Truecrypt partition early in case you often suffer from shutdown hangs (if you do you may find you are unable to mount the Truecrypt partition under Windows until you successfully boot and exit Linux).

```
#!/bin/sh
#
# Copyright (c) 2002 SuSE Linux AG Nuernberg, Germany. All rights reserved.
#
# Author: Werner Fink <werner@suse.de>, 1998
#         Burchard Steinbild, 1998
#
# /etc/init.d/halt.local
#
# script with local commands to be executed from init on system shutdown
#
# Here you should add things, that should happen directly before shutting
# down.
#
#Here you unmount the Truecrypt partition in case of system crash on Shutdown
truecrypt -d /dev/sda8

#Here you are making the encrypted swap plain again.
swapoff /dev/mapper/swap
cryptsetup remove swap
mkswap /dev/sda6
```

H. Example hibernation script file

The following script is named '04swapencryption' and can be found in the directory '/etc/pm/sleep.d'. The highlighted sections should be edited to match your system. The unused sections 'suspend' and 'resume' are used during suspending to RAM.

```
#!/bin/bash
case $1 in
  hibernate)
    echo "Unmounting encrypted swap, plain formatting and mounting basic swap"
    swapoff /dev/mapper/swap
    cryptsetup remove swap
    mkswap /dev/sda6
    swapon /dev/sda6
    ;;
  suspend)
    echo ""
    ;;
  thaw)
    echo "Re-encrypting, formatting and mounting swap"
    swapoff /dev/sda6
    cryptsetup -c blowfish -h sha256 -d /dev/urandom create swap /dev/sda6
    mkswap /dev/mapper/swap
    swapon /dev/mapper/swap
    ;;
  resume)
    echo ""
    ;;
  *)
    echo ""
    ;;
esac
```

I. Example SwapFs Registry Entry

The following is an example of the SwapFS registry entry used on this system. The green highlighted sections show how the partition was set to the correct swap partition location (in this system /dev/sda6 = Partition 3), and the blue shows how the drive letter was changed to 'Y:' from 'S:'.

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SwapFs]
```

```
"ErrorControl"=dword:00000001
```

```
"Group"="Filter"
```

```
#
```

```
# When to start the driver:
```

```
#   At boot:   Start=1
```

```
#   Manually: Start=3
```

```
#
```

```
"Start"=dword:00000001
```

```
"Type"=dword:00000001
```

```
#
```

```
# (/dev/hda1 in Linux = \\Device\\Harddisk0\\Partition1 in NT, an extended
```

```
# partition is skipped in the enumeration)
```

```
#
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SwapFs\Parameters]
```

```
"SwapDevice"="\\Device\\Harddisk0\\Partition3"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\DOS Devices]
```

```
"Y:"="\\Device\\Harddisk0\\Partition3"
```

J. /etc/suspend.conf example

The following is an example edited suspend.conf script. The highlighted sections are the edits made to the default file.

This example covers 2 pages in this howto.

```
#####  
##  
## note:  
## using pm-utils or powersaved, this file (/etc/suspend.conf) only serves as  
## a template, image size and resume device are filled in dynamically  
## and the generated /var/lib/s2disk.conf is used to suspend.  
## _If_ you enter stuff here, it will be copied to that file unchanged,  
## but this might skip some features and sanity checks.  
##  
#####  
##  
## your snapshot device. You should not need to change this.  
# snapshot device = /dev/snapshot  
#  
## enter your swap device here. Read the warning on pm-utils above, please!  
#resume device = <path to resume device file>  
#  
## image size will also be filled in by pm-utils  
#image size = 350000000  
#  
#suspend loglevel = 2  
#max loglevel =  
#  
## compute checksum will slow down suspend and resume. Debugging option  
#compute checksum = y  
#  
## compression will often speed up suspend and resume (default y)  
#compress = n  
#  
## encryption support is rather basic right now - e.g. USB keyboards will not  
## work to enter the key in the standard initrd, also beware of  
## non-US keyboard layouts. Only use this if you know what you are doing.  
encrypt = y
```

```
#
## RSA key file that is used for encryption
RSA key file = /etc/suspend.key
#
#
#early writeout = n
#splash = y
#
## shutdown method:
## platform - go through ACPI BIOS to power off the machine (default)
## shutdown - just power off like after a shutdown
## reboot - reboot instead of powering off. For debugging only.
#shutdown method = platform
#
## resume offset: for use with swapfiles, use "swap-offset" to find out.
#resume offset = 12345
#
```