TRUECRYPT

FREE OPEN-SOURCE ON-THE-FLY ENCRYPTION

USER'S GUIDE

www.truecrypt.org

Version Information

TrueCrypt User's Guide, version 6.1a. Released December 1, 2008.

Licensing Information

By installing, running, using, copying, (re)distributing, and/or modifying TrueCrypt or a portion thereof you accept all terms, responsibilities and obligations contained in the TrueCrypt License the full text of which is contained in the file *License.txt* included in TrueCrypt binary and source code distribution packages.

Copyright Information

Trademark Information

Limitations

The TrueCrypt Foundation does not warrant that the information contained in this document meets your requirements or that it is free of errors.

CONTENTS

PREFACE

Please note that although most chapters of this documentation apply generally to all versions of TrueCrypt, some sections are primarily aimed at users of the Windows versions of TrueCrypt. Hence, such sections may contain information that is inappropriate in regards to the Mac OS X and Linux versions of TrueCrypt.

Introduction

TrueCrypt is a software system for establishing and maintaining an on-the-fly-encrypted volume (data storage device). On-the-fly encryption means that data are automatically encrypted or decrypted right before they are loaded or saved, without any user intervention. No data stored on an encrypted volume can be read (decrypted) without using the correct password/keyfile(s) or correct encryption keys. Entire file system is encrypted (e.g., file names, folder names, contents of every file, free space, meta data, etc).

Files can be copied to and from a mounted TrueCrypt volume just like they are copied to/from any normal disk (for example, by simple drag-and-drop operations). Files are automatically being decrypted on-the-fly (in memory/RAM) while they are being read or copied from an encrypted TrueCrypt volume. Similarly, files that are being written or copied to the TrueCrypt volume are automatically being encrypted on-the-fly (right before they are written to the disk) in RAM. Note that this does *not* mean that the *whole* file that is to be encrypted/decrypted must be stored in RAM before it can be encrypted/decrypted. There are no extra memory (RAM) requirements for TrueCrypt. For an illustration of how this is accomplished, see the following paragraph.

Let's suppose that there is an .avi video file stored on a TrueCrypt volume (therefore, the video file is entirely encrypted). The user provides the correct password (and/or keyfile) and mounts (opens) the TrueCrypt volume. When the user double clicks the icon of the video file, the operating system launches the application associated with the file type – typically a media player. The media player then begins loading a small initial portion of the video file from the TrueCrypt-encrypted volume to RAM (memory) in order to play it. While the portion is being loaded, TrueCrypt is automatically decrypting it (in RAM). The decrypted portion of the video (stored in RAM) is then played by the media player. While this portion is being played, the media player begins loading next small portion of the video file from the TrueCrypt-encrypted volume to RAM (memory) and the process repeats. This process is called on-the-fly encryption/decryption and it works for all file types, not only for video files.

Note that TrueCrypt never saves any decrypted data to a disk – it only stores them temporarily in RAM (memory). Even when the volume is mounted, data stored in the volume is still encrypted. When you restart Windows or turn off your computer, the volume will be dismounted and files stored in it will be inaccessible (and encrypted). Even when power supply is suddenly interrupted (without proper system shut down), files stored in the volume are inaccessible (and encrypted). To make them accessible again, you have to mount the volume (and provide the correct password and/or keyfile).

Beginner's Tutorial

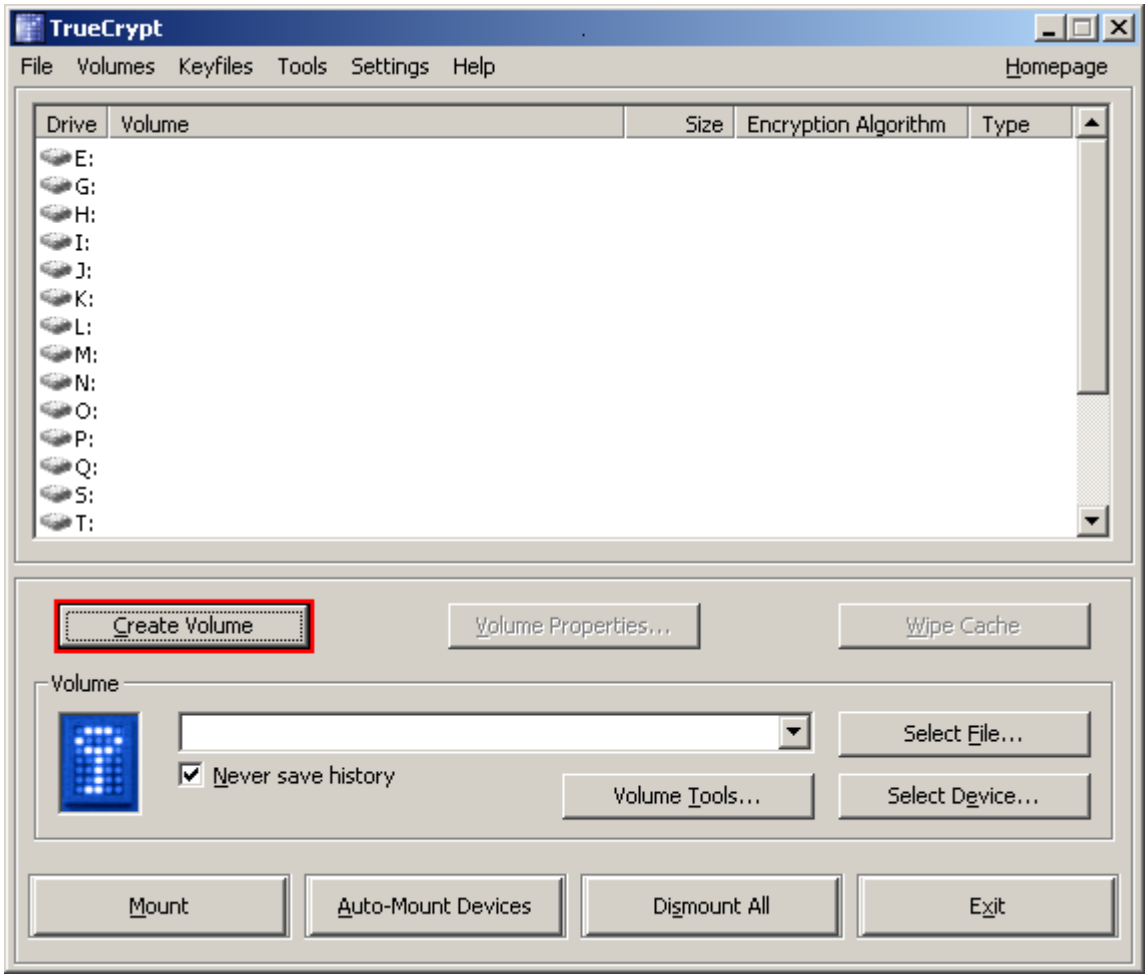
How to Create and Use a TrueCrypt Container

This chapter contains step-by-step instructions on how to create, mount, and use a TrueCrypt volume. We strongly recommend that you also read the other sections of this manual, as they contain important information.

STEP 1:

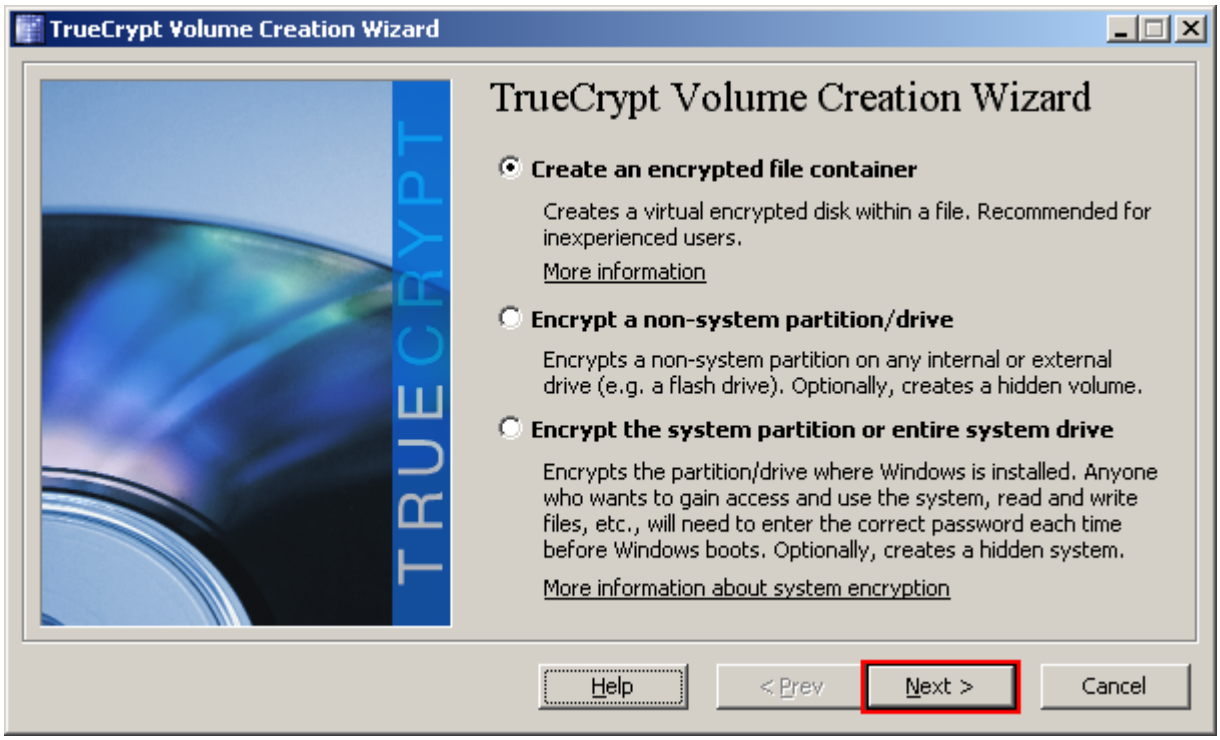
If you have not done so, download and install TrueCrypt. Then launch TrueCrypt by double-clicking the file TrueCrypt.exe or by clicking the TrueCrypt shortcut in your Windows Start menu.

STEP 2:



The main TrueCrypt window should appear. Click **Create Volume** (marked with a red rectangle for clarity).

STEP 3:



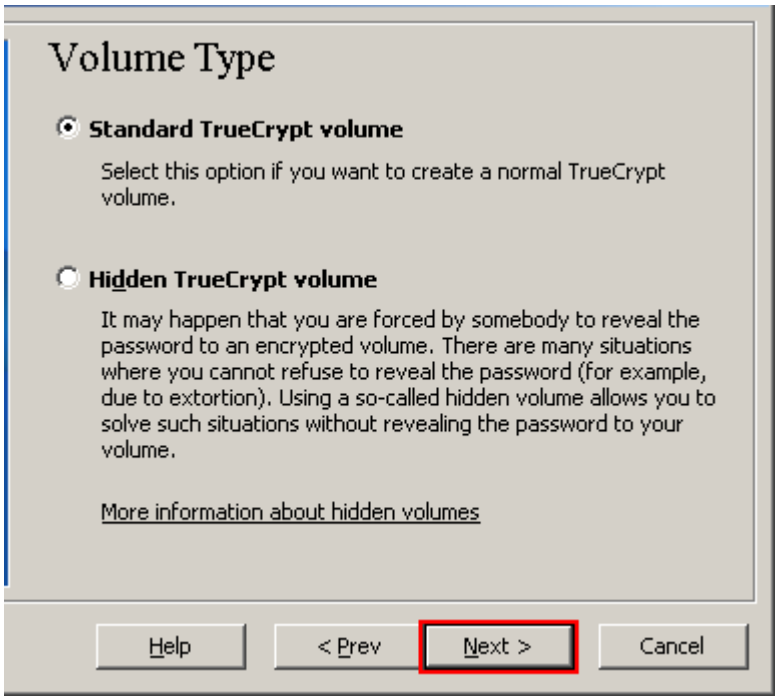
The TrueCrypt Volume Creation Wizard window should appear.

In this step you need to choose where you wish the TrueCrypt volume to be created. A TrueCrypt volume can reside in a file, which is also called container, in a partition or drive. In this tutorial, we will choose the first option and create a TrueCrypt volume within a file.

As the option is selected by default, you can just click **Next**.

Note: In the following steps the screenshots will show only the right-hand part of the Wizard window.

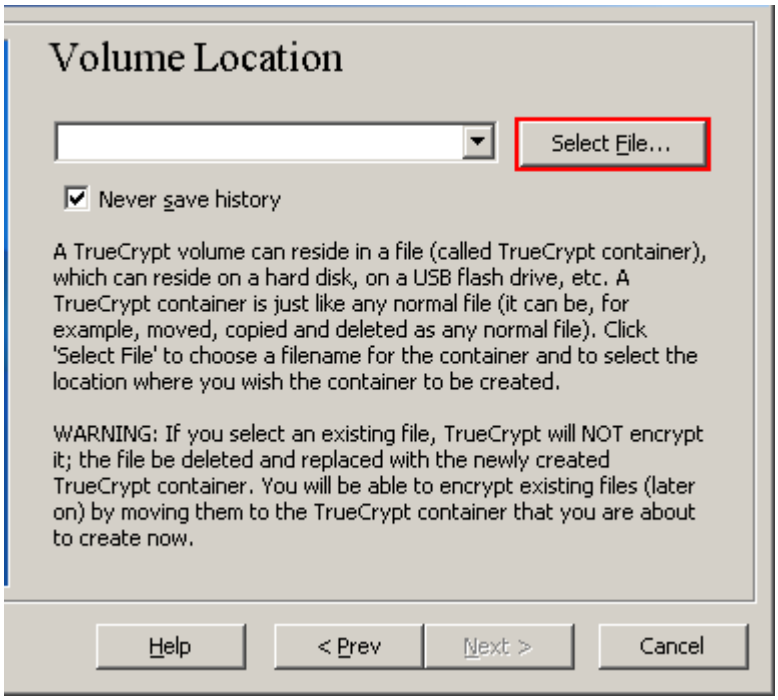
STEP 4:



In this step you need to choose whether to create a standard or hidden TrueCrypt volume. In this tutorial, we will choose the former option and create a standard TrueCrypt volume.

As the option is selected by default, you can just click **Next**.

STEP 5:

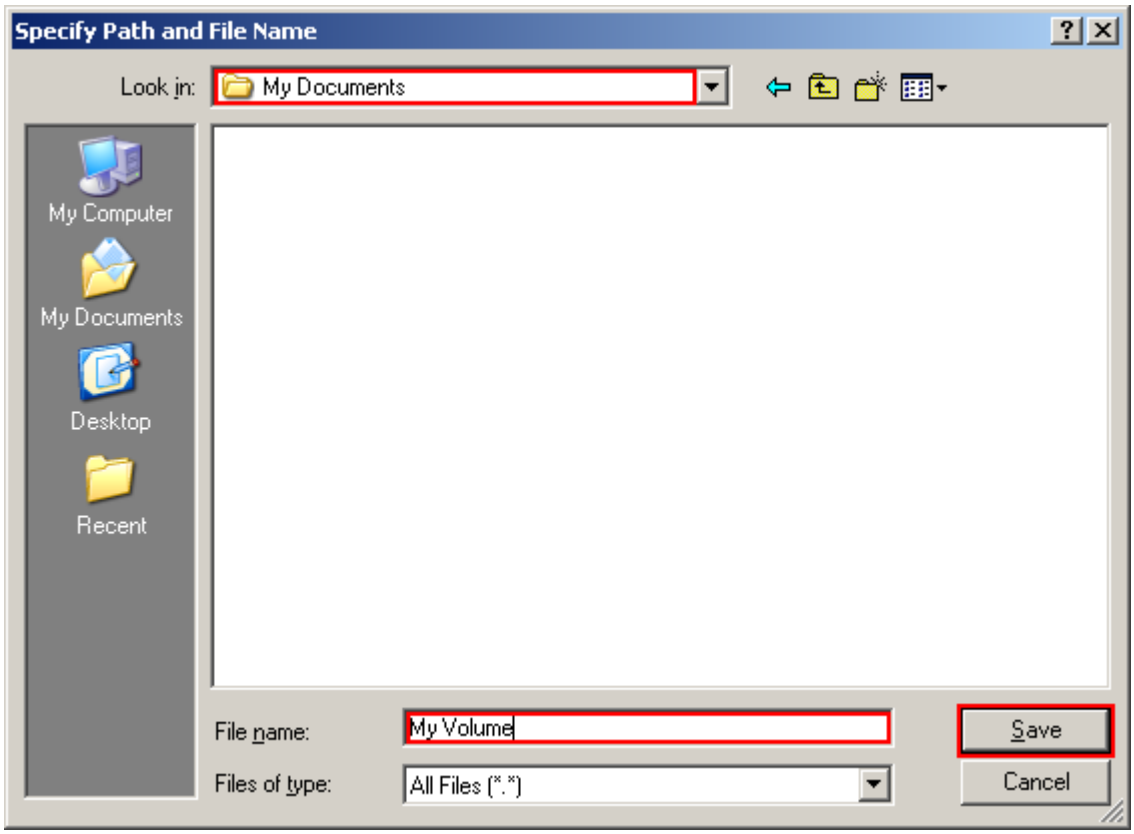


In this step you have to specify where you wish the TrueCrypt volume (file container) to be created. Note that a TrueCrypt container is just like any normal file. It can be moved, copied and deleted as any normal file. It also needs a filename, which you will choose in the next step.

Click **Select File**.

The standard Windows file selector should appear (while the window of the TrueCrypt Volume Creation Wizard remains open in the background).

STEP 6:



In this tutorial, we will create our TrueCrypt volume in the folder *D:\My Documents*\ and the filename of the volume (container) will be *My Volume* (as can be seen in the screenshot above). You may, of course, choose any other filename and location you like (for example, on a USB memory stick). Note that the file *My Volume* does not exist yet – TrueCrypt will create it.

IMPORTANT: Note that TrueCrypt will *not* encrypt any existing files. If you select an existing file, it will be overwritten and replaced by the newly created volume (so the overwritten file will be *lost*, *not* encrypted). You will be able to encrypt existing files (later on) by moving them to the TrueCrypt volume that we are creating now.[*]

Select the desired path (where you wish the container to be created) in the file selector.

Type the desired container filename in the **File name** box.

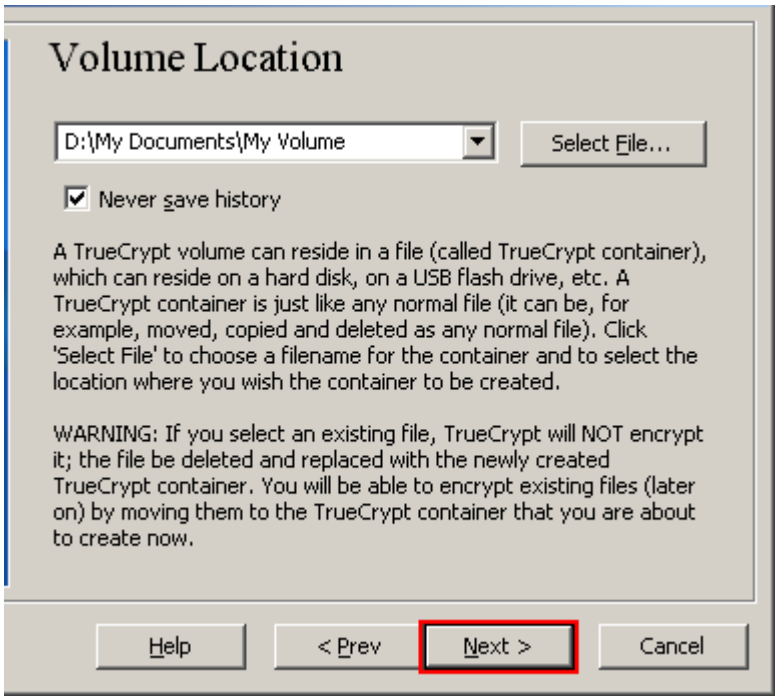
Click **Save**.

The file selector window should disappear.

In the following steps, we will return to the TrueCrypt Volume Creation Wizard.

[*] Note that after you copy existing unencrypted files to a TrueCrypt volume, you should securely erase (wipe) the original unecrypted files. There are software tools that can be used for the purpose of secure erasure (many of them are free).

STEP 7:

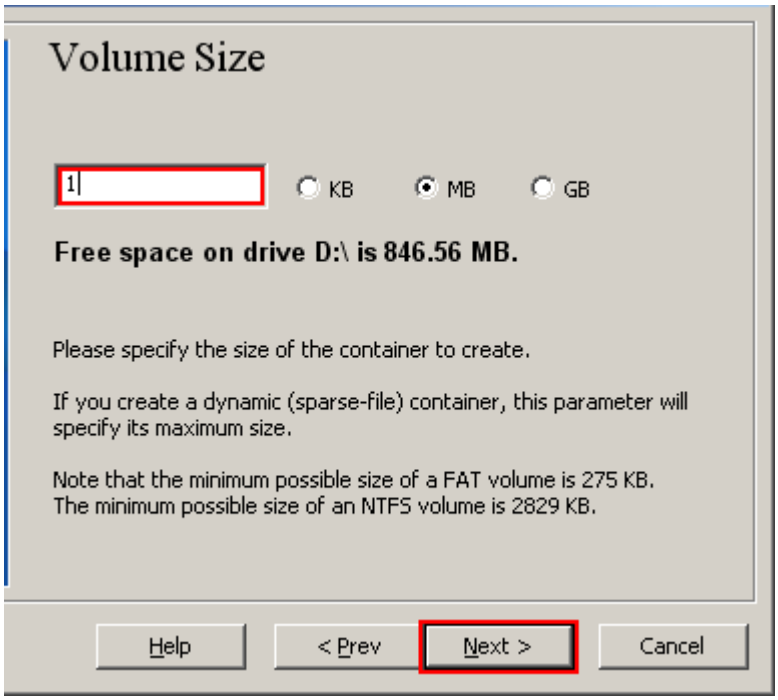


In the Volume Creation Wizard window, click **Next**.

STEP 8:

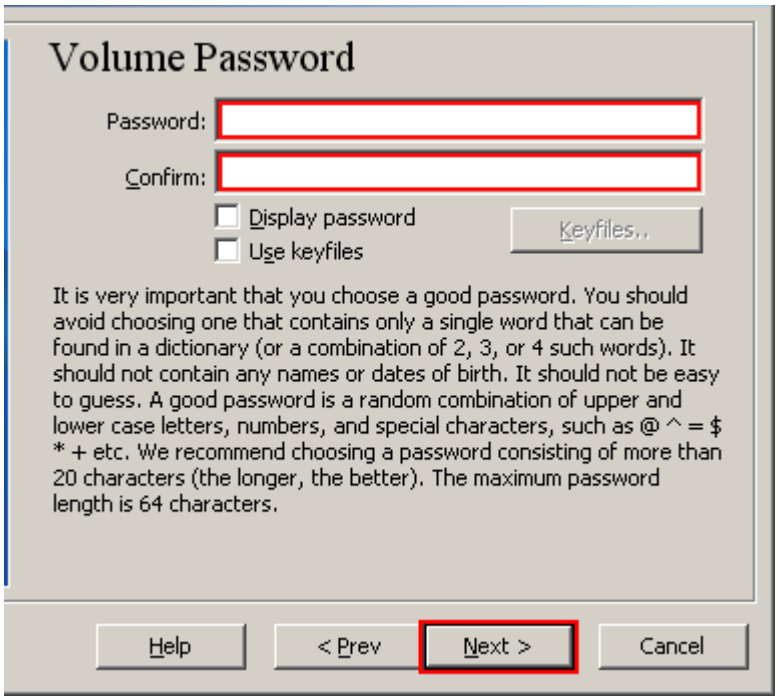
Here you can choose an encryption algorithm and a hash algorithm for the volume. If you are not sure what to select here, you can use the default settings and click **Next** (for more information, see Chapters *Encryption Algorithms* and *Hash Algorithms*).

STEP 9:



Here we specify that we wish the size of our TrueCrypt container to be 1 megabyte. You may, of course, specify a different size. After you type the desired size in the input field (marked with a red rectangle), click **Next**.

STEP 10:



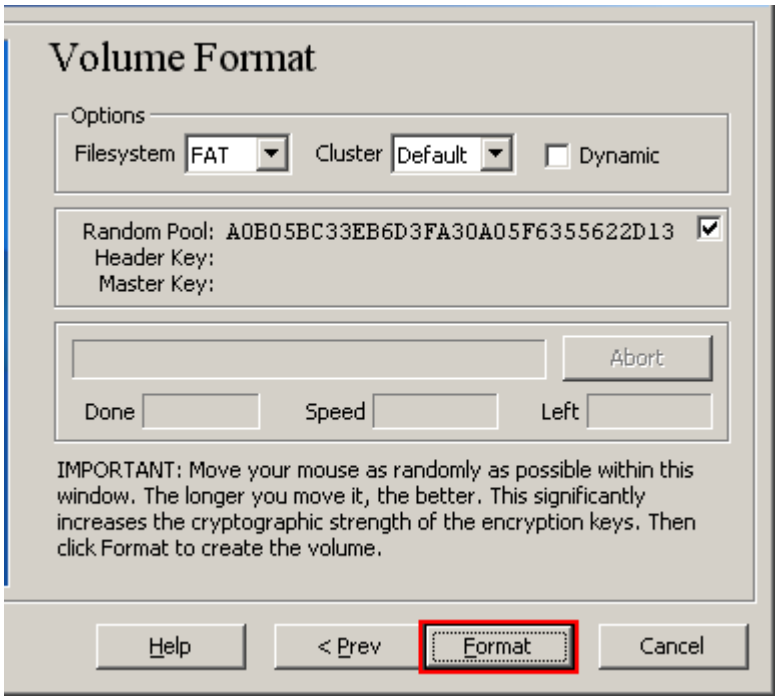
This is one of the most important steps. Here you have to choose a good volume password.

Read carefully the information displayed in the Wizard window about what is considered a good password.

After you choose a good password, type it in the first input field. Then re-type it in the input field below the first one and click **Next**.

Note: The button **Next** will be disabled until passwords in both input fields are the same.

STEP 11:



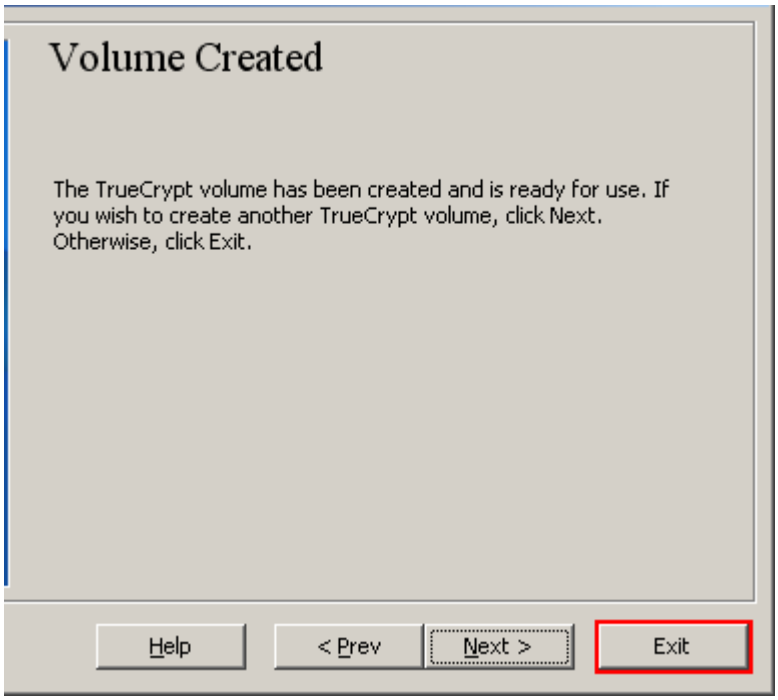
Move your mouse as randomly as possible within the Volume Creation Wizard window at least for 30 seconds. The longer you move the mouse, the better. This significantly increases the cryptographic strength of the encryption keys (which increases security).

Click **Format**.

Volume creation should begin. TrueCrypt will now create a file called *My Volume* in the folder *D:\My Documents* (as we specified in Step 6). This file will be a TrueCrypt container (it will contain the encrypted TrueCrypt volume). Depending on the size of the volume, the volume creation may take a long time. After it finishes, the following dialog box will appear:

Click **OK** to close the dialog box.

STEP 12:



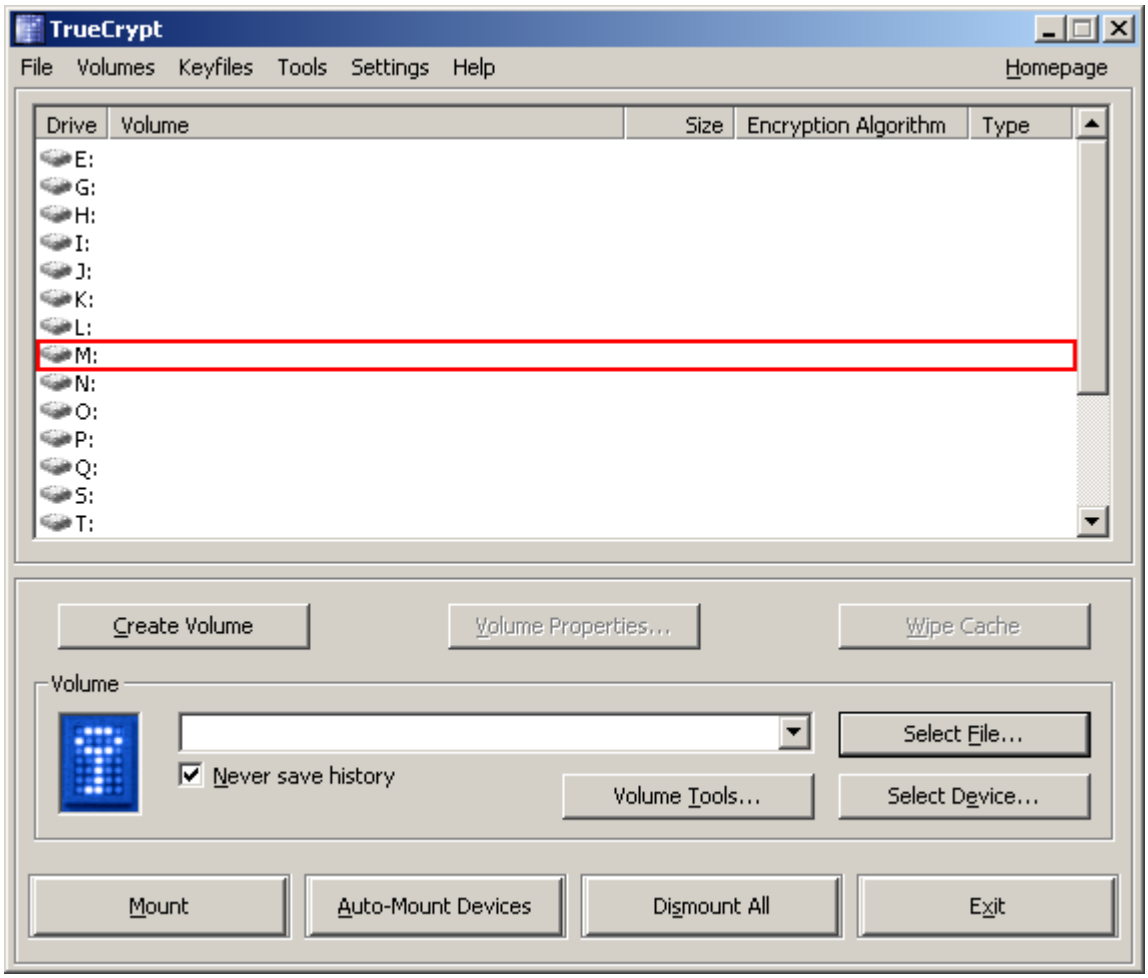
We have just successfully created a TrueCrypt volume (file container).

In the TrueCrypt Volume Creation Wizard window, click **Exit**.

The Wizard window should disappear.

In the remaining steps, we will mount the volume we just created. We will return to the main TrueCrypt window (which should still be open, but if it is not, repeat Step 1 to launch TrueCrypt and then continue from Step 13.)

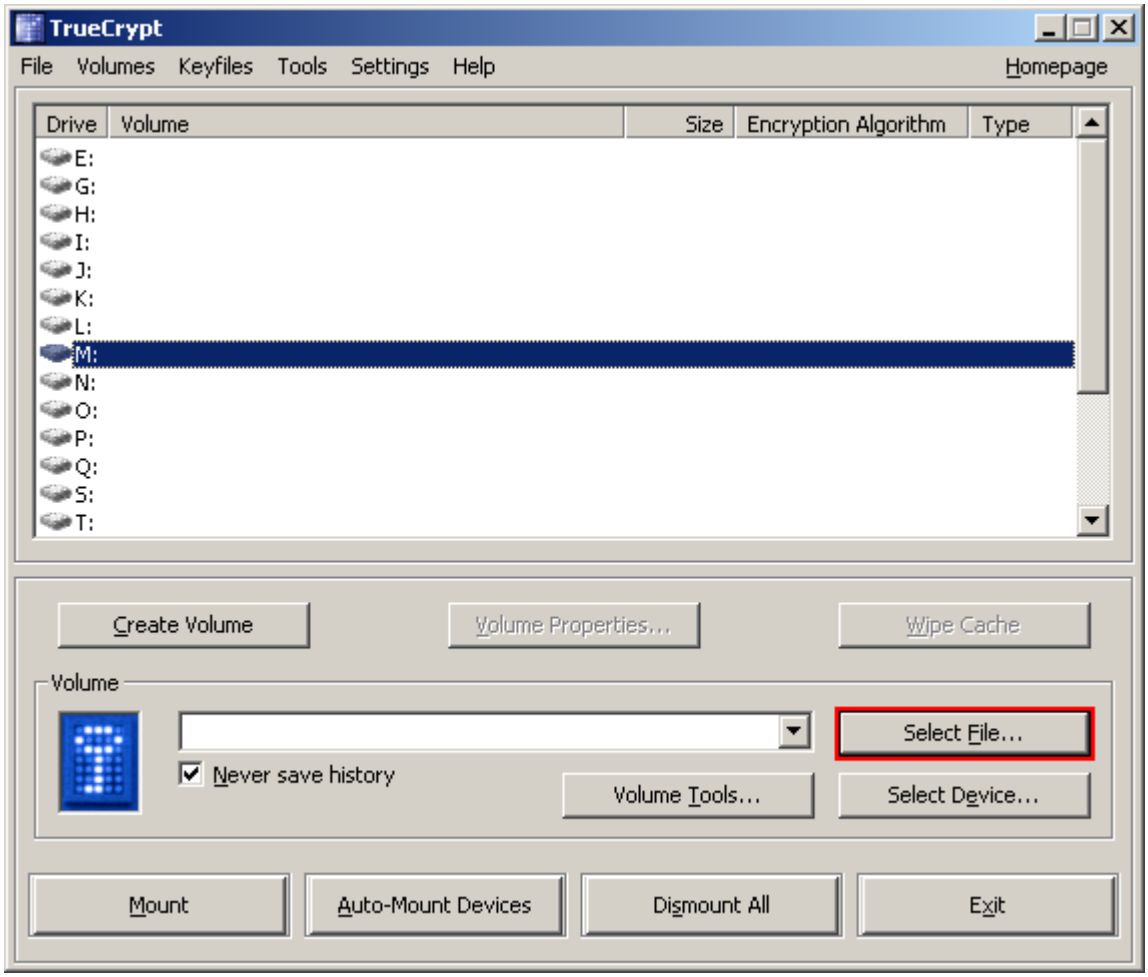
STEP 13:



Select a drive letter from the list (marked with a red rectangle). This will be the drive letter to which the TrueCrypt container will be mounted.

Note: In this tutorial, we chose the drive letter M, but you may of course choose any other available drive letter.

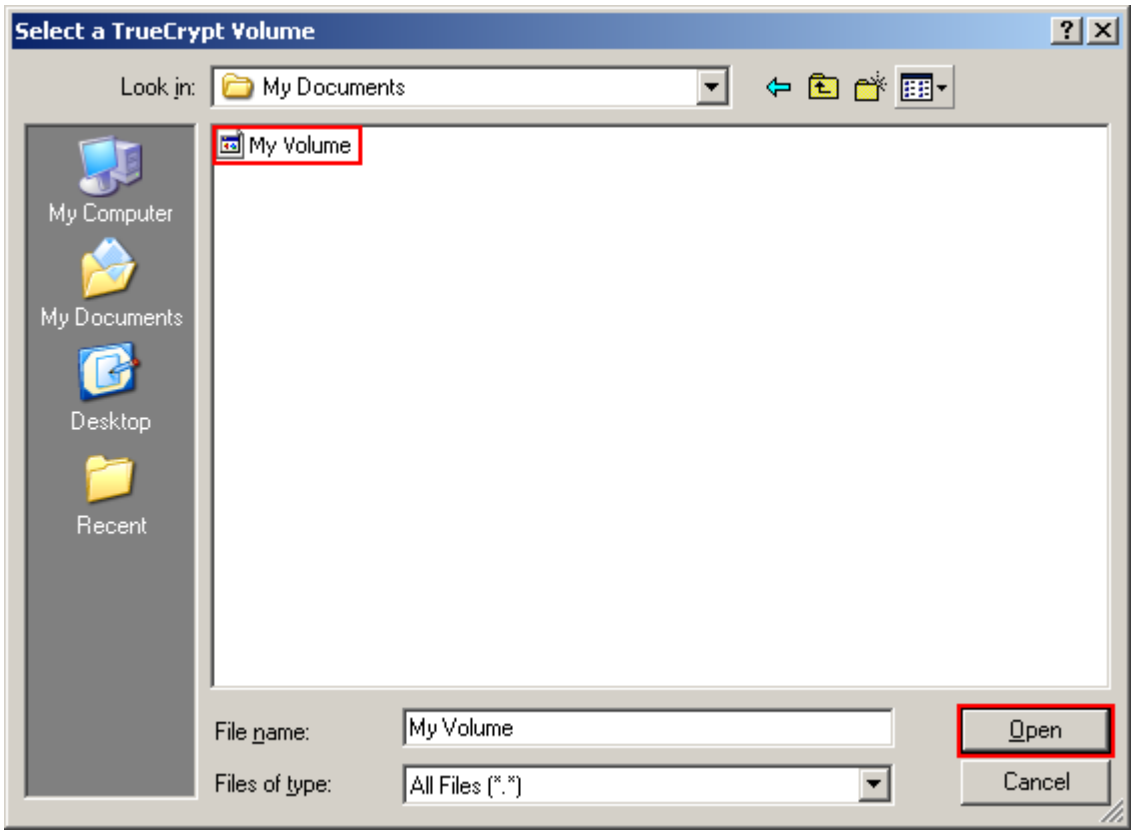
STEP 14:



Click **Select File**.

The standard file selector window should appear.

STEP 15:



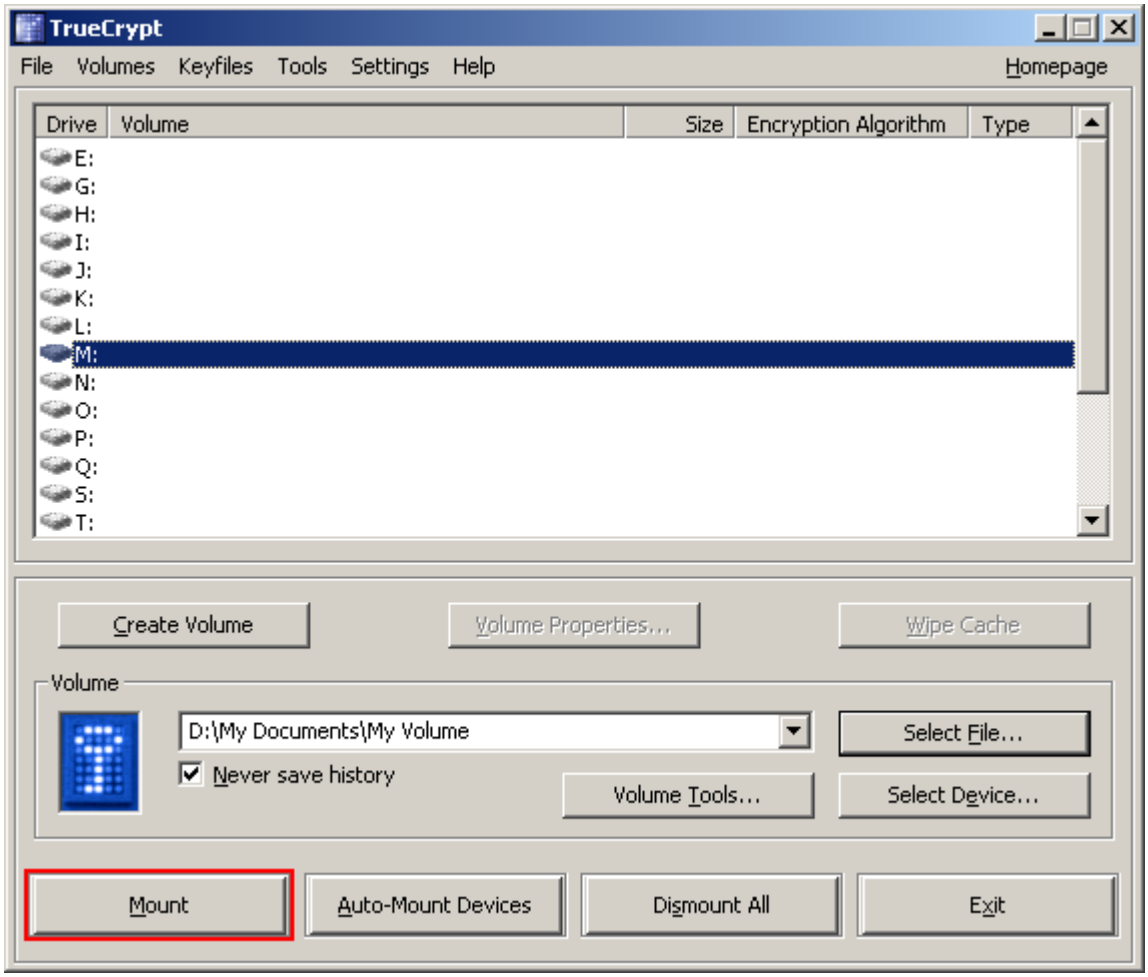
In the file selector, browse to the container file (which we created in Steps 6-11) and select it.

Click **Open** (in the file selector window).

The file selector window should disappear.

In the following steps, we will return to the main TrueCrypt window.

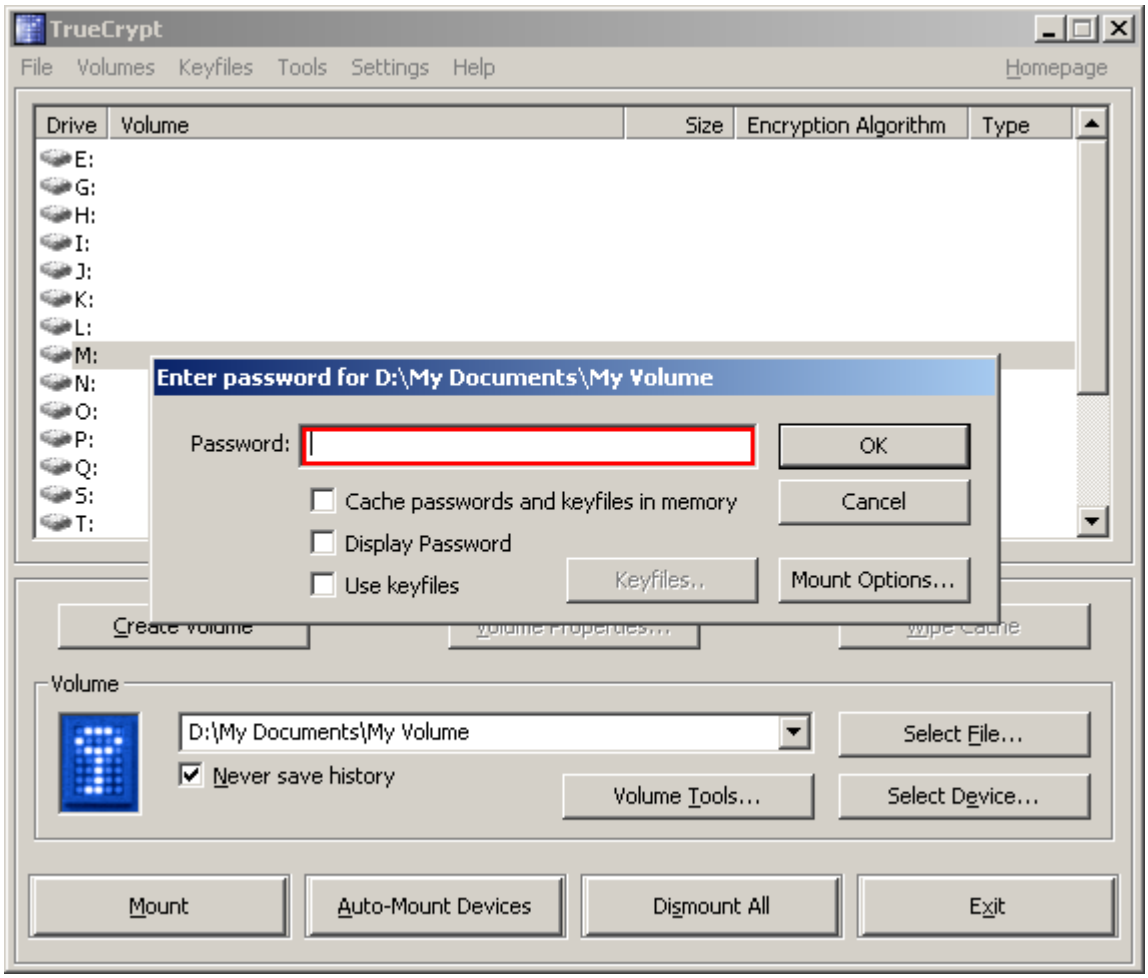
STEP 16:



In the main TrueCrypt window, click **Mount**.

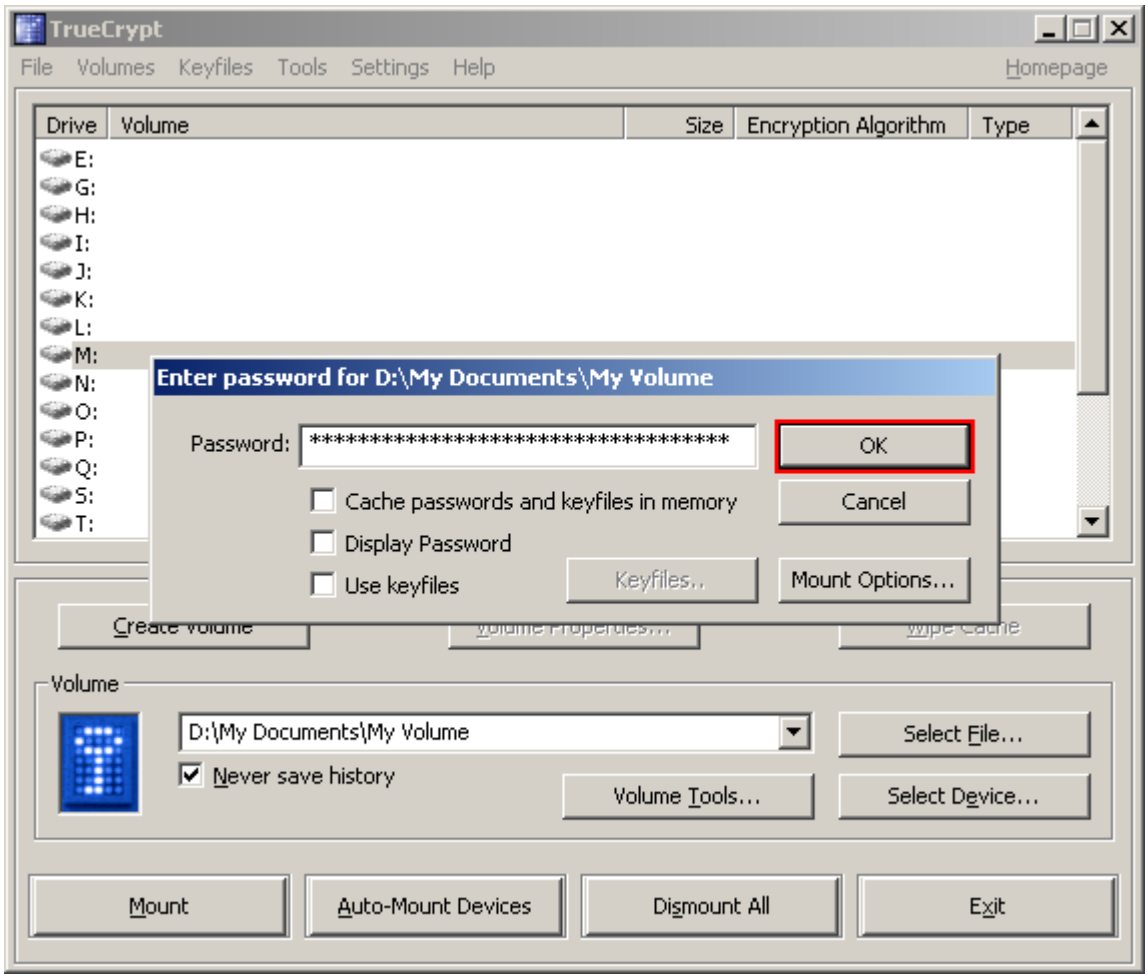
Password prompt dialog window should appear.

STEP 17:



Type the password (which you specified in Step 10) in the password input field (marked with a red rectangle).

STEP 18:

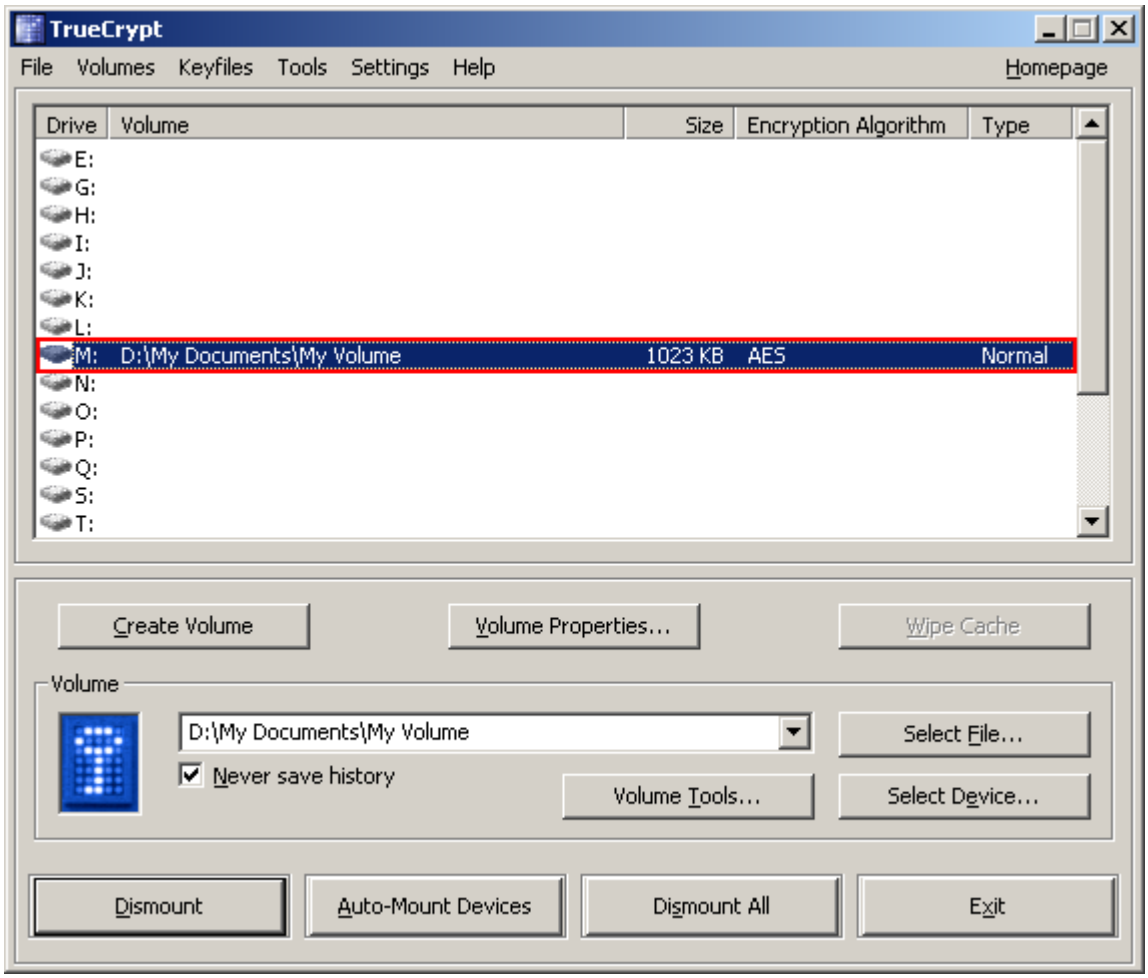


Click **OK** in the password prompt window.

TrueCrypt will now attempt to mount the volume. If the password is incorrect (for example, if you typed it incorrectly), TrueCrypt will notify you and you will need to repeat the previous step (type the password again and click **OK**). If the password is correct, the volume will be mounted.

(Continued on the next page.)

FINAL STEP:



We have just successfully mounted the container as a virtual disk M:

The virtual disk is entirely encrypted (including file names, allocation tables, free space, etc.) and behaves like a real disk. You can save (or copy, move, etc.) files to this virtual disk and they will be encrypted on-the-fly as they are being written.

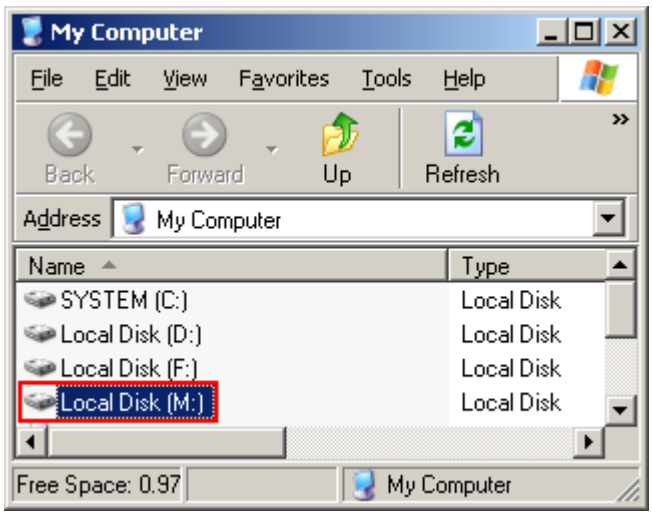
If you open a file stored on a TrueCrypt volume, for example, in media player, the file will be automatically decrypted to RAM (memory) on-the-fly while it is being read.

Important: Note that when you open a file stored on a TrueCrypt volume (or when you write/copy a file to/from the TrueCrypt volume) you will not be asked to enter the password again. You need to enter the correct password only when mounting the volume.

You can open the mounted volume, for example, by double-clicking the item marked with a red rectangle in the screenshot above.

(Continued on the next page.)

You can also browse to the mounted volume the way you normally browse to any other types of volumes. For example, by opening the '*Computer*' (or '*My Computer*') list and double clicking the corresponding drive letter (in this case, it is the letter M).

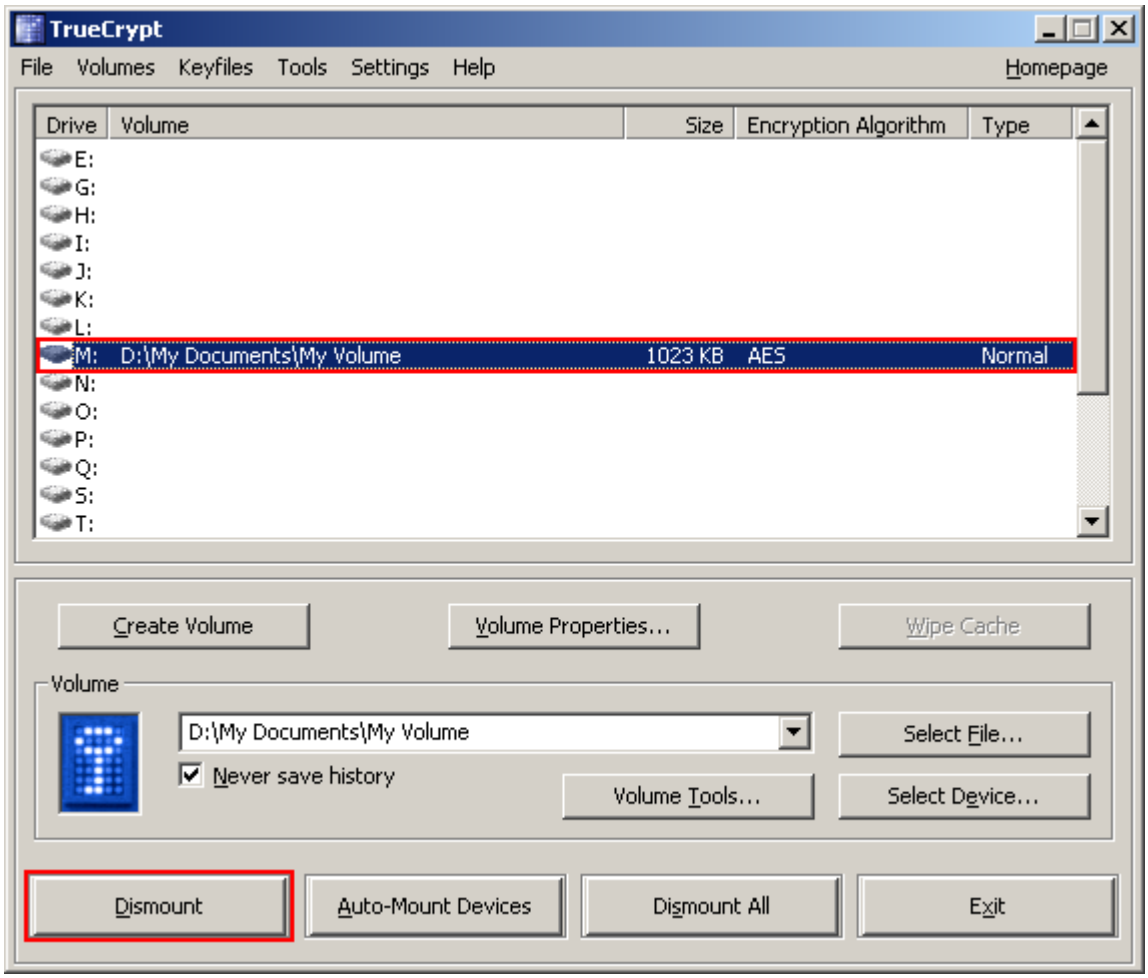


You can copy files to and from the TrueCrypt volume just as you would copy them to any normal disk (for example, by simple drag-and-drop operations). Files that are being read or copied from the encrypted TrueCrypt volume are automatically decrypted on-the-fly (in memory/RAM). Similarly, files that are being written or copied to the encrypted TrueCrypt volume are automatically encrypted on-the-fly (right before they are written to the disk) in RAM.

Note that TrueCrypt never saves any decrypted data to a disk – it only stores them temporarily in RAM (memory). Even when the volume is mounted, data stored in the volume is still encrypted. When you restart Windows or turn off your computer, the volume will be dismounted and all files stored on it will be inaccessible (and encrypted). Even when power supply is suddenly interrupted (without proper system shut down), all files stored on the volume will be inaccessible (and encrypted). To make them accessible again, you have to mount the volume. To do so, repeat Steps 13-18.

(Continued on the next page.)

If you want to close the volume and make files stored on it inaccessible, either restart your operating system or dismount the volume. To do so, follow these steps:



Select the volume from the list of mounted volumes in the main TrueCrypt window (marked with a red rectangle in the screenshot above) and then click **Dismount** (also marked with a red rectangle in the screenshot above). To make files stored on the volume accessible again, you will have to mount the volume. To do so, repeat Steps 13-18.

How to Create and Use a TrueCrypt Partition/Device

Instead of creating file containers, you can also encrypt physical partitions or drives (i.e., create TrueCrypt device-hosted volumes). To do so, repeat the steps 1-3 but in the step 3 select the second or third option. Then follow the remaining instructions in the wizard. When you create a device-hosted TrueCrypt volume within a *non-system* partition/drive, you can mount it by clicking *Auto-Mount Devices* in the main TrueCrypt window. For information pertaining to encrypted *system* partition/drives, see the chapter *System Encryption*.

Important: ***We strongly recommend that you also read the other chapters of this manual, as they contain important information that has been omitted in this tutorial for simplicity.***

Plausible Deniability

In case an adversary forces you to reveal your password, TrueCrypt provides and supports two kinds of plausible deniability:

1. Hidden volumes (for more information, see the section *Hidden Volume* below).

2. It is impossible to identify a TrueCrypt volume. Until decrypted, a TrueCrypt volume appears to consist of nothing more than random data (it does not contain any kind of "signature"). Therefore, it is impossible to *prove* that a file, a partition or a device is a TrueCrypt volume or that it has been encrypted. However, note that for system encryption, the first drive track contains the (unencrypted) TrueCrypt Boot Loader, which can be easily identified as such (for more information, see the chapter *System Encryption*). In such cases, plausible deniability can be achieved by creating a hidden operating system (see the section *Hidden Operating System*).

TrueCrypt containers (file-hosted volumes) can have any file extension you like (for example, .raw, .iso, .img, .dat, .rnd, .tc) or they can have no file extension at all. TrueCrypt ignores file extensions. If you need plausible deniability, make sure your TrueCrypt volumes do not have the .tc file extension (this file extension is officially associated with TrueCrypt).

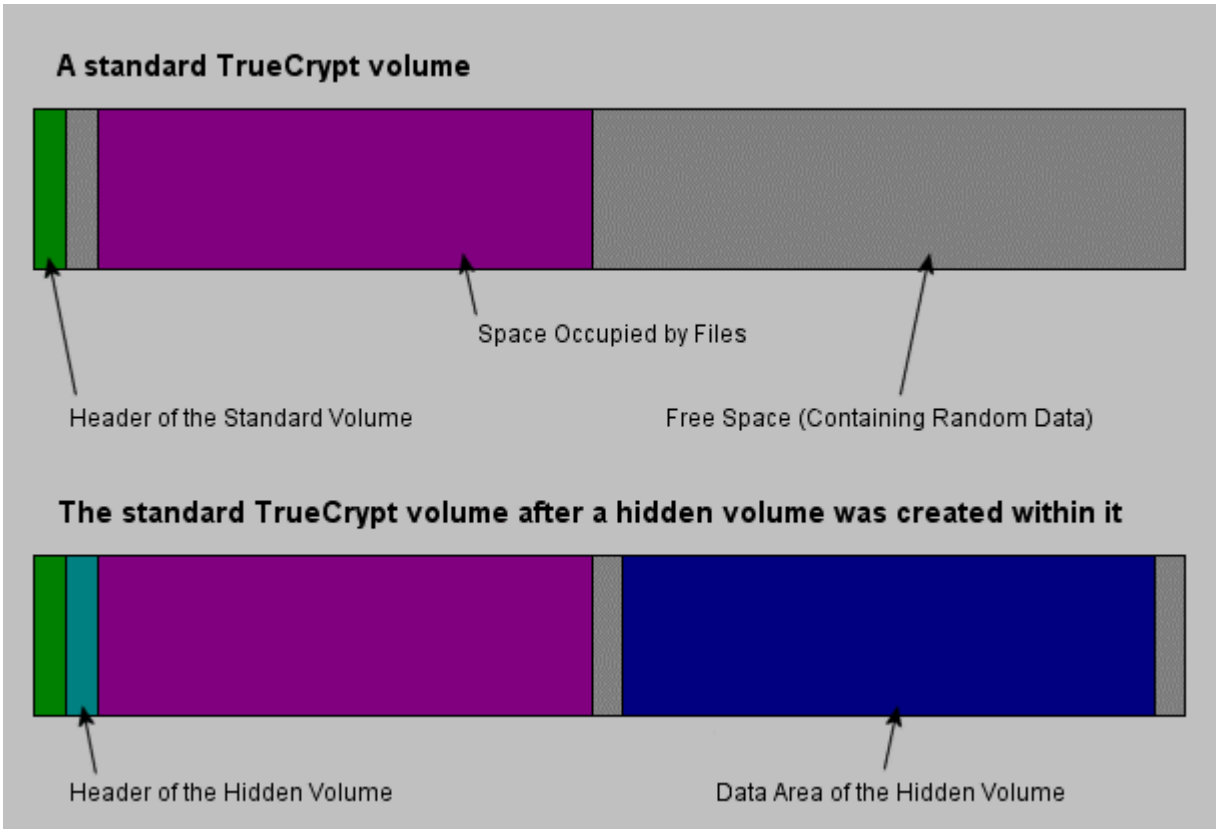
When formatting a hard disk partition as a TrueCrypt volume, the partition table (including the partition type) is *never* modified (no TrueCrypt “signature” or “ID” is written to the partition table).

Whenever TrueCrypt accesses a file-hosted volume (e.g., when dismounting, attempting to mount, changing or attempting to change the password, creating a hidden volume within it, etc.) or a keyfile, it preserves the timestamp of the container/keyfile (i.e., date and time that the container/keyfile was last accessed* or last modified), unless this behavior is disabled in the preferences.

* Note that if you use the Windows ‘File Properties’ tool to view a container/keyfile timestamp (e.g., by right-clicking the container/keyfile and selecting ‘Properties’), you will alter the date and time that the container/keyfile was last *accessed*. Also note that if you view thumbnails of files in the Windows file selector (for instance, when selecting a container or keyfile in the Thumbnail file selector mode), Windows may modify the timestamps of the files (date and time that the files were last accessed).

Hidden Volume

It may happen that you are forced by somebody to reveal the password to an encrypted volume. There are many situations where you cannot refuse to reveal the password (for example, due to extortion). Using a so-called hidden volume allows you to solve such situations without revealing the password to your volume.



The layout of a standard TrueCrypt volume before and after a hidden volume was created within it.

The principle is that a TrueCrypt volume is created within another TrueCrypt volume (within the free space on the volume). Even when the outer volume is mounted, it is impossible to prove whether there is a hidden volume within it or not[*], because free space on *any* TrueCrypt volume is always filled with random data when the volume is created[†] and no part of the (dismounted) hidden volume can be distinguished from random data. Note that TrueCrypt does not modify the file system (information about free space, etc.) within the outer volume in any way.

The password for the hidden volume must be substantially different from the password for the outer volume. To the outer volume, (before creating the hidden volume within it) you should copy some sensitive-looking files that you actually do NOT want to hide. These files will be there for

[*] Provided that all the instructions in the TrueCrypt Volume Creation Wizard have been followed and provided that the precautions mentioned in the subsection '*Security Precautions Pertaining to Hidden Volumes*' are followed.
† Provided that the options *Quick Format* and *Dynamic* are disabled and provided that the volume does not contain a filesystem that has been encrypted in place (TrueCrypt does not allow the user to create a hidden volume within such a volume). For information on the method used to fill free volume space with random data, see chapter *Technical Details*, section *TrueCrypt Volume Format Specification.*

anyone who would force you to hand over the password. You will reveal only the password for the outer volume, not for the hidden one. Files that really are sensitive will be stored on the hidden volume.

A hidden volume can be mounted the same way as a standard TrueCrypt volume: Click *Select File* or *Select Device* to select the outer/host volume (important: make sure the volume is *not* mounted). Then click *Mount*, and enter the password for the hidden volume. Whether the hidden or the outer volume will be mounted is determined by the entered password (i.e., when you enter the password for the outer volume, then the outer volume will be mounted; when you enter the password for the hidden volume, the hidden volume will be mounted).

TrueCrypt first attempts to decrypt the standard volume header using the entered password. If it fails, it loads the area of the volume where a hidden volume header can be stored (i.e. the bytes 65536–131071, which contain solely random data when there is no hidden volume within the volume) to RAM and attempts to decrypt it using the entered password. Note that hidden volume headers cannot be identified, as they appear to consist entirely of random data. If the header is successfully decrypted (for information on how TrueCrypt determines that it was successfully decrypted, see the section *Encryption Scheme*), the information about the size of the hidden volume is retrieved from the decrypted header (which is still stored in RAM), and the hidden volume is mounted (its size also determines its offset).

A hidden volume can be created within any type of TrueCrypt volume, i.e., within a file-hosted volume or partition/device-hosted volume (requires administrator privileges). To create a hidden TrueCrypt volume, click on *Create Volume* in the main program window and select *Create a hidden TrueCrypt volume*. The Wizard will provide help and all information necessary to successfully create a hidden TrueCrypt volume.
When creating a hidden volume, it may be very difficult or even impossible for an inexperienced user to set the size of the hidden volume such that the hidden volume does not overwrite data on the outer volume. Therefore, the Volume Creation Wizard automatically scans the cluster bitmap of the outer volume (before the hidden volume is created within it) and determines the maximum possible size of the hidden volume.*

If there are any problems when creating a hidden volume, refer to the chapter *Troubleshooting* for possible solutions.

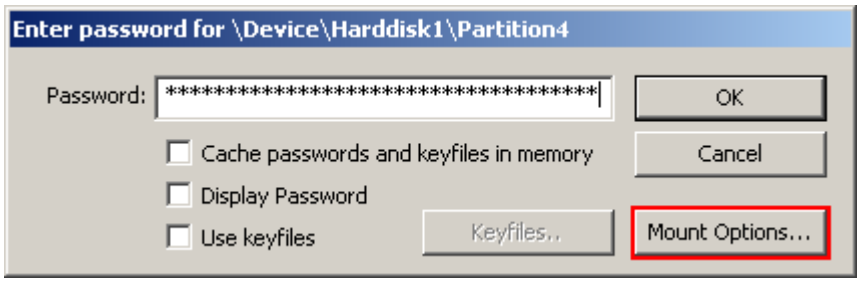
Note that it is also possible to create and boot an operating system residing in a hidden volume (see the section *Hidden Operating System* in the chapter *Plausible Deniability*).

* The wizard scans the cluster bitmap to determine the size of the uninterrupted area of free space (if there is any) whose end is aligned with the end of the outer volume. This area accommodates the hidden volume and therefore the size of this area limits the maximum possible size of the hidden volume. On Linux and Mac OS X, the wizard actually does not scan the cluster bitmap, but the driver detects any data written to the outer volume and uses their position as previously described.

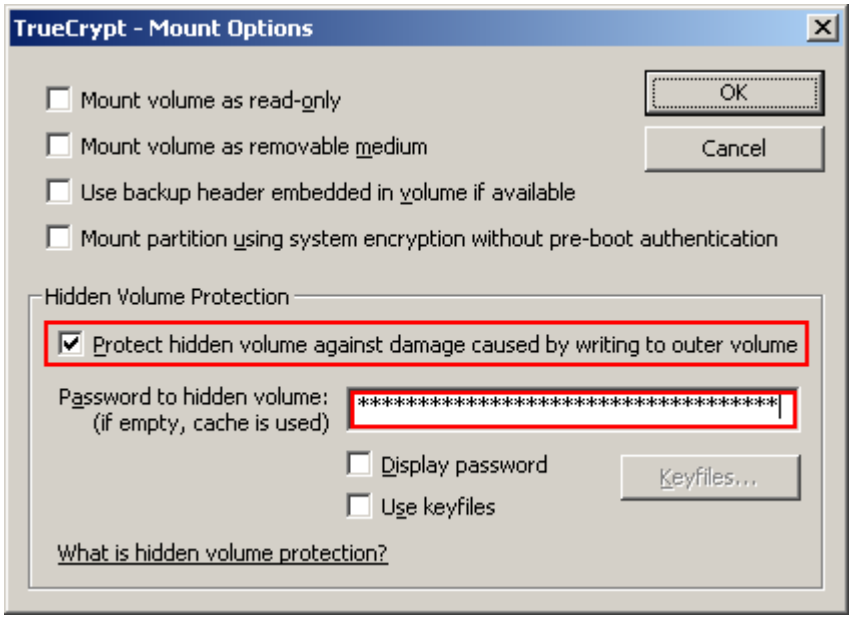
Protection of Hidden Volumes Against Damage

If you mount a TrueCrypt volume within which there is a hidden volume, you may *read* data stored on the (outer) volume without any risk. However, if you (or the operating system) need to *save* data to the outer volume, there is a risk that the hidden volume will get damaged (overwritten). To prevent this, you should protect the hidden volume in a way described in this section.

When mounting an outer volume, type in its password and before clicking *OK,* click *Mount Options*:



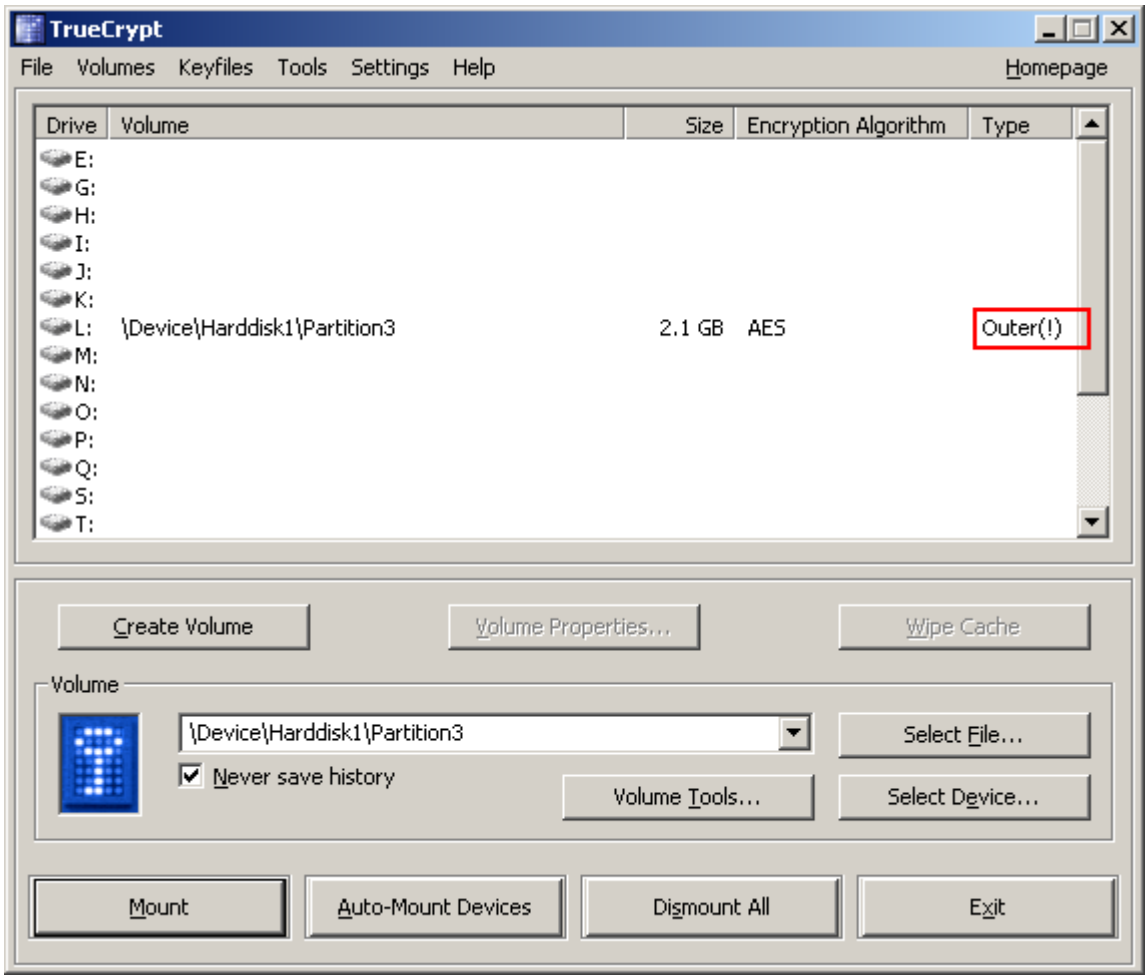
In the *Mount Options* dialog window, enable the option '*Protect hidden volume against damage caused by writing to outer volume* '. In the '*Password to hidden volume*' input field, type the password for the hidden volume. Click *OK* and in the main password entry dialog click *OK*.



Both passwords must be correct; otherwise, the outer volume will not be mounted. When hidden volume protection is enabled, TrueCrypt does *not* actually mount the hidden volume. It only decrypts its header (in RAM) and retrieves information about the size of the hidden volume (from the decrypted header). Then, the outer volume is mounted and any attempt to save data to the area of the hidden volume will be rejected (until the outer volume is dismounted). **Note that TrueCrypt *never* modifies the filesystem (e.g., information about allocated clusters, amount of free space, etc.) within the outer volume in any way. As soon as the volume is dismounted, the protection is lost. When the volume is mounted again, it is *not* possible to determine whether the volume has used hidden volume protection or not. The hidden**

volume protection can be activated only by users who supply the correct password (and/or keyfiles) for the hidden volume (each time they mount the outer volume).

As soon as a write operation to the hidden volume area is denied/prevented (to protect the hidden volume), the entire host volume (both the outer and the hidden volume) becomes write-protected until dismounted (the TrueCrypt driver reports the 'invalid parameter' error to the system upon each attempt to write data to the volume). This preserves plausible deniability (otherwise certain kinds of inconsistency within the file system could indicate that this volume has used hidden volume protection). When damage to hidden volume is prevented, a warning is displayed (provided that the TrueCrypt Background Task is enabled – see the chapter *TrueCrypt Background Task*). Furthermore, the type of the mounted outer volume displayed in the main window changes to '*Outer(!)*':



Moreover, the field *Hidden Volume Protected* in the *Volume Properties* dialog window says: '*Yes (damage prevented!)*'.

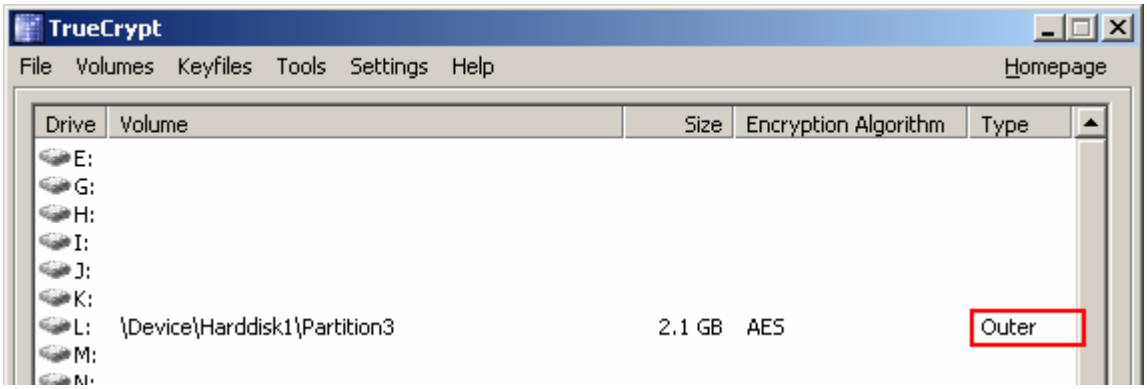
Note that when damage to hidden volume is prevented, *no* information about the event is written to the volume. When the outer volume is dismounted and mounted again, the volume properties will *not* display the string "*damage prevented*".

There are several ways to check that a hidden volume is being protected against damage:

1. A confirmation message box saying that hidden volume is being protected is displayed after the outer volume is mounted (if it is not displayed, the hidden volume is not protected!).

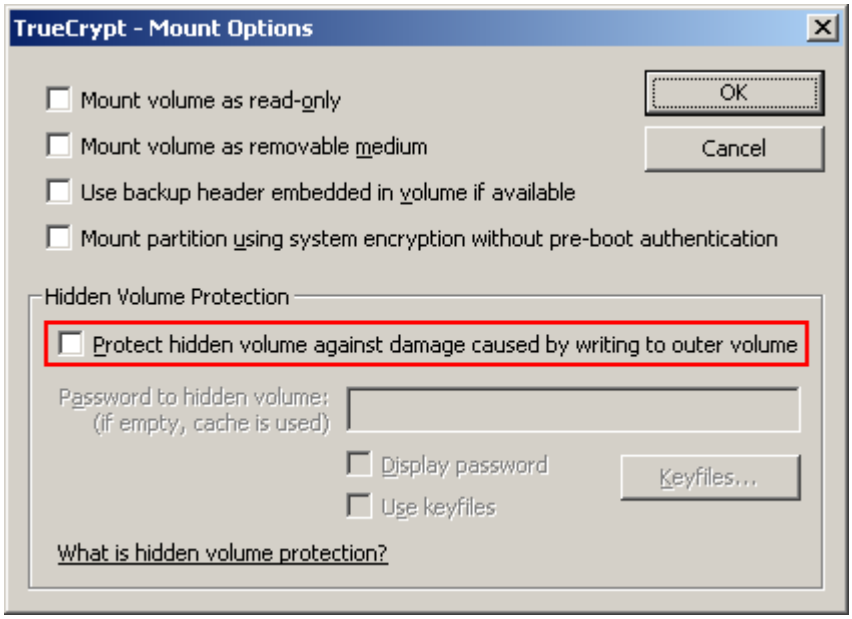
2. In the *Volume Properties* dialog, the field *Hidden Volume Protected* says '*Yes*':

3. The type of the mounted outer volume is *Outer*:



Important: When an adversary asks you to mount an outer volume, you, of course, must not mount the outer volume with the hidden volume protection enabled. Note that during the time when an outer volume is mounted with the hidden volume protection enabled, the adversary can find out that a hidden volume exists within the outer volume (he/she will be able to find it out until the volume is dismounted).

Warning: Note that the option '*Protect hidden volume against damage caused by writing to outer volume*' in the *Mount Options* dialog window is automatically disabled after a mount attempt is completed, no matter whether it is successful or not (all hidden volumes that are already being protected will, of course, continue to be protected). Therefore, you need to check that option *each* time you attempt to mount the outer volume (if you wish the hidden volume to be protected):



If you want to mount an outer volume and protect a hidden volume within using cached passwords, then follow these steps: Hold down the *Control* (*Ctrl*) key when clicking *Mount* (or select *Mount with Options* from the *Volumes* menu). This will open the *Mount Options* dialog. Enable the option '*Protect hidden volume against damage caused by writing to outer volume*' and leave the password box empty. Then click *OK*.

If you need to mount an outer volume and you know that you will not need to save any data to it, then the most comfortable way of protecting the hidden volume against damage is mounting the outer volume as read-only (see the section *Mount Options*).

Security Precautions Pertaining to Hidden Volumes

If you use a hidden TrueCrypt volume, you must follow these security precautions:

- If an adversary has access to a (dismounted) TrueCrypt volume at several points over time, he may be able to determine which sectors of the volume are changing. If you change the contents of a hidden volume (e.g., create/copy new files to the hidden volume or modify/delete/rename/move files stored on the hidden volume, etc.), the contents of sectors (ciphertext) in the hidden volume area will change. After being given the password to the outer volume, the adversary might demand an explanation why these sectors changed. Your failure to provide a plausible explanation might indicate the existence of a hidden volume within the outer volume.

  Note that the issue described above may also arise, for example, in the following cases:

  - The file system in which you store a file-hosted TrueCrypt container has been defragmented and a copy of the TrueCrypt container (or of its fragment) remains in the free space on the host volume (in the defragmented file system). To prevent this, do one of the following:
    - Use a partition/device-hosted TrueCrypt volume instead of file-hosted.
    - Securely erase free space on the host volume (in the defragmented file system) after defragmenting.
    - Do not defragment file systems in which you store TrueCrypt volumes.

  - A file-hosted TrueCrypt container is stored in a journaling file system (such as NTFS). A copy of the TrueCrypt container (or of its fragment) may remain on the host volume. To prevent this, do one the following:
    - Use a partition/device-hosted TrueCrypt volume instead of file-hosted.
    - Store the container in a non-journaling file system (for example, FAT32).

  - A TrueCrypt volume resides on a device, which utilizes a wear-leveling mechanism (e.g. some USB flash drives). A copy of (a fragment of) the TrueCrypt volume may remain on the device. For more information on wear-leveling, see the section *Wear-Leveling* in the chapter *Security Precautions.*

- Make sure that *Quick Format* is disabled when encrypting a partition/device within which you intend to create a hidden volume.

- On Windows, make sure you have not deleted any files within a volume within which you intend to create a hidden volume (the cluster bitmap scanner does not detect deleted files).

- On Linux or Mac OS X, if you intend to create a hidden volume within a file-hosted TrueCrypt volume, make sure that the volume is not sparse-file-hosted (the Windows version of TrueCrypt verifies this and disallows creation of hidden volumes within sparse files).

- When a hidden volume is *mounted*, the operating system and third-party applications may write to non-hidden volumes (typically, to the unencrypted system volume) unencrypted information about the data stored in the hidden volume (e.g. filenames and locations of recently accessed files, databases created by file indexing tools, etc.), or the data itself in an unencrypted form (temporary files, etc.), or unencrypted information about the filesystem residing in the hidden volume (which might be used e.g. to identify the filesystem and to determine whether it is the filesystem residing in the outer volume). Therefore, the following guidelines and precautions must be followed:

    - *Windows*: Create a hidden operating system (for information on how to do so, see the section *Hidden Operating System*) and mount hidden volumes only when the hidden operating system is running.

      Note: When a hidden operating system is running, TrueCrypt ensures that all local unencrypted filesystems and non-hidden TrueCrypt volumes are read-only (i.e. no files can be written to such filesystems or TrueCrypt volumes).[*] Data is allowed to be written to filesystems within hidden TrueCrypt volumes.

    - *Linux*: Download or create a "live CD" version of your Linux operating system (i.e. a "live" Linux system entirely stored on and booted from a CD/DVD) that ensures that any data written to the system volume is written to a RAM disk. Mount hidden volumes only when such a "live CD" system is running. During the session, only filesystems that reside in hidden TrueCrypt volumes may be mounted in read-write mode (outer or unencrypted volumes/filesystems must be mounted as read-only or must not be mounted/accessible at all). If you cannot use such a "live CD" version of the operating system or if you are not able to ensure that applications and the standard version (as opposed to a "live CD" version) of your operating system do not write the above types of sensitive data to non-hidden volumes (or filesystems), you should not mount or create hidden TrueCrypt volumes under Linux.

    - *Mac OS X*: If you are not able to ensure that applications and the operating system do not write the above types of sensitive data to non-hidden volumes (or filesystems), you should not mount or create hidden TrueCrypt volumes under Mac OS X.

- If you use an **operating system residing within a hidden volume** (see the section *Hidden Operating System*), then, in addition to the above precautions, you must follow these security precautions:

    - You should use the decoy operating system as frequently as you use your computer. Ideally, you should use it for all activities that do not involve sensitive data. Otherwise, plausible deniability of the hidden operating system might be adversely affected (if you revealed the password for the decoy operating system to an adversary, he could find out that the system is not used very often, which might indicate the existence of a hidden operating system on your computer). Note that you can save data to the decoy system partition anytime without any risk that the hidden volume will get damaged (because the decoy system is not installed in the outer volume).

[*] This does not apply to filesystems on CD/DVD-like media and on custom, untypical, or non-standard devices/media.

- If the operating system requires activation, it must be activated before it is cloned (cloning is part of the process of creation of a hidden operating system — see the section *Hidden Operating System*) and the hidden operating system (i.e. the clone) must never be reactivated. The reason is that the hidden operating system is created by copying the content of the system partition to a hidden volume (so if the operating system is not activated, the hidden operating system will not be activated either). If you activated or reactivated a hidden operating system, the date and time of the activation (and other data) might be logged on a Microsoft server (and on the hidden operating system) but not on the decoy operating system. Therefore, if an adversary had access to the data stored on the server or intercepted your request to the server (and if you revealed the password for the decoy operating system to him), he might find out that the decoy operating system was activated (or reactivated) at a different time, which might indicate the existence of a hidden operating system on your computer.

  For similar reasons, any software that requires activation must be installed and activated before you start creating the hidden operating system.

- When you need to shut down the hidden system and start the decoy system, do *not* restart the computer. Instead, shut it down or hibernate it and then leave it powered off for several minutes before turning the computer on and booting the decoy system. This is required to clear the memory, which may contain sensitive data. For more information, see the section *Unencrypted Data in RAM* in the chapter *Security Precautions*.

- The computer may be connected to a network (including the internet) only when the decoy operating system is running. When the hidden operating system is running, the computer should not be connected to any network, including the internet (one of the most reliable ways to ensure it is to unplug the network cable, if there is one). Note that if data is downloaded from or uploaded to a remote server, the date and time of the connection, and other data, are typically logged on the server. Various kinds of data are also logged on the operating system (e.g. Windows auto-update data, application logs, error logs, etc.) Therefore, if an adversary had access to the data stored on the server or intercepted your request to the server (and if you revealed the password for the decoy operating system to him), he might find out that the connection was not made from within the decoy operating system, which might indicate the existence of a hidden operating system on your computer.

  Also note that similar issues would affect you if there were any filesystem shared over a network under the hidden operating system (regardless of whether the filesystem is remote or local). Therefore, when the hidden operating system is running, there must be no filesystem shared over a network (in any direction).

- If the BIOS, EFI, or any other component logs power-down events or any other events that could be related to events logged in Windows logs, you should either disable such logging or ensure that the log is securely erased after each session.

In addition to the above precautions, you must follow the security precautions listed in the following chapters:

- *Security Precautions*
- *How to Back Up Securely*

Hidden Operating System

If your system partition or system drive is encrypted using TrueCrypt, you need to enter your pre-boot authentication password in the TrueCrypt Boot Loader screen after you turn on or restart your computer. It may happen that you are forced by somebody to decrypt the operating system or to reveal the pre-boot authentication password. There are many situations where you cannot refuse to do so (for example, due to extortion). TrueCrypt allows you to create a hidden operating system whose existence will be impossible to prove (provided that certain guidelines are followed — see below). Thus, you will not have to decrypt or reveal the password for the hidden operating system.

Before you continue reading this section, make sure you have read the section ***Hidden Volume*** and that you understand what a hidden TrueCrypt volume is.

A **hidden operating system** is a system (for example, Windows Vista or Windows XP) that is installed in a hidden TrueCrypt volume. It is impossible to prove that a hidden TrueCrypt volume exists (provided that certain guidelines are followed; for more information, see the section *Hidden Volume*) and, therefore, it is impossible to prove that a hidden operating system exists.

However, in order to boot a system encrypted by TrueCrypt, an unencrypted copy of the TrueCrypt Boot Loader has to be stored on the system drive or on a TrueCrypt Rescue Disk. Hence, the mere presence of the TrueCrypt Boot Loader can indicate that there is a system encrypted by TrueCrypt on the computer. Therefore, to provide a plausible explanation for the presence of the TrueCrypt Boot Loader, the TrueCrypt wizard helps you create a second encrypted operating system, so-called **decoy operating system**, during the process of creation of a hidden operating system. A decoy operating system must not contain any sensitive files. Its existence is not secret (it is *not* installed in a hidden volume). The password for the decoy operating system can be safely revealed to anyone forcing you to disclose your pre-boot authentication password.[*]

You should use the decoy operating system as frequently as you use your computer. Ideally, you should use it for all activities that do not involve sensitive data. Otherwise, plausible deniability of the hidden operating system might be adversely affected (if you revealed the password for the decoy operating system to an adversary, he could find out that the system is not used very often, which might indicate the existence of a hidden operating system on your computer). Note that you can save data to the decoy system partition anytime without any risk that the hidden volume will get damaged (because the decoy system is *not* installed in the outer volume — see below).

There will be two pre-boot authentication passwords — one for the hidden system and the other for the decoy system. If you want to start the hidden system, you simply enter the password for the hidden system in the TrueCrypt Boot Loader screen (which appears after you turn on or restart your computer). Likewise, if you want to start the decoy system (for example, when asked to do so by an adversary), you just enter the password for the decoy system in the TrueCrypt Boot Loader screen.

Note: When you enter a pre-boot authentication password, the TrueCrypt Boot Loader first attempts to decrypt (using the entered password) the last 512 bytes of the first logical track of the system drive (where encrypted master key data for non-hidden encrypted system partitions/drives are normally stored). If it fails and if there is a partition behind the boot partition, the TrueCrypt Boot

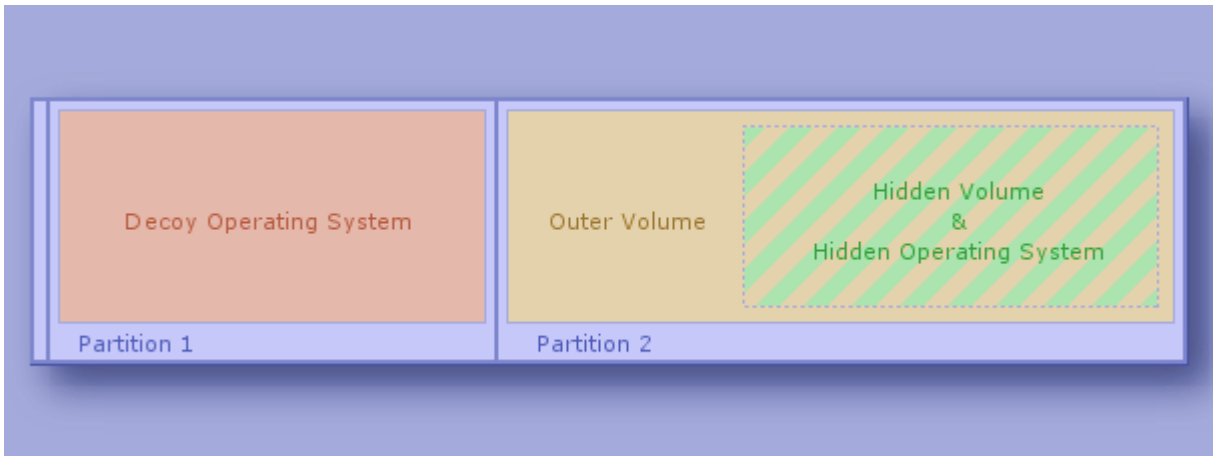
[*] It is not practical (and therefore is not supported) to install operating systems in two TrueCrypt volumes that are embedded within a single partition, because using the outer operating system would often require data to be written to the area of the hidden operating system (and if such write operations were prevented using the hidden volume protection feature, it would inherently cause system crashes, i.e. 'Blue Screen' errors).

Loader (even if there is actually no hidden volume on the drive) automatically tries to decrypt (using the same entered password again) the area of the first partition behind the boot partition where the encrypted header of a possible hidden volume might be stored. Note that TrueCrypt never knows if there is a hidden volume in advance (the hidden volume header cannot be identified, as it appears to consist entirely of random data). If the header is successfully decrypted (for information on how TrueCrypt determines that it was successfully decrypted, see the section *Encryption Scheme*), the information about the size of the hidden volume is retrieved from the decrypted header (which is still stored in RAM), and the hidden volume is mounted (its size also determines its offset). For further technical details, see the section *Encryption Scheme* in the chapter *Technical Details*.

When running, the hidden operating system appears to be installed on the same partition as the original operating system (the decoy system). However, in reality, it is installed within the partition behind it (in a hidden volume). All read/write operations are transparently redirected from the system partition to the hidden volume. Neither the operating system nor applications will know that data written to and read from the system partition is actually written to and read from the partition behind it (from/to a hidden volume). Any such data is encrypted and decrypted on the fly as usual (with an encryption key different from the one that is used for the decoy operating system).

Note that there will also be a third password — the one for the **outer volume**. It is not a pre-boot authentication password, but a regular TrueCrypt volume password. It can be safely disclosed to anyone forcing you to reveal the password for the encrypted partition where the hidden volume (containing the hidden operating system) resides. Thus, the existence of the hidden volume (and of the hidden operating system) will remain secret. If you are not sure you understand how this is possible, or what an outer volume is, please read the section *Hidden Volume*. The outer volume should contain some sensitive-looking files that you actually do *not* want to hide.

To summarize, there will be three passwords in total. Two of them can be revealed to an attacker (for the decoy system and for the outer volume). The third password, for the hidden system, must remain secret.



Example Layout of System Drive Containing Hidden Operating System

Process of Creation of Hidden Operating System

To start the process of creation of a hidden operating system, select *System* > *Create Hidden Operating System* and then follow the instructions in the wizard.

Initially, the wizard verifies that there is a suitable partition for a hidden operating system on the system drive. Note that before you can create a hidden operating system, you need to create a partition for it on the system drive. It must be the first partition behind the system partition and it must be at least 5% larger than the system partition (the system partition is the one where the currently running operating system is installed). However, if the outer volume (not to be confused with the system partition) is formatted as NTFS, the partition for the hidden operating system must be at least 110% (2.1 times) larger than the system partition (the reason is that the NTFS file system always stores internal data exactly in the middle of the volume and, therefore, the hidden volume, which is to contain a clone of the system partition, can reside only in the second half of the partition).

In the next steps, the wizard will create two TrueCrypt volumes (outer and hidden) within the first partition behind the system partition. The hidden volume will contain the hidden operating system. The size of the hidden volume is always the same as the size of the system partition. The reason is that the hidden volume will need to contain a clone of the content of the system partition (see below). Note that the clone will be encrypted using a different encryption key than the original. Before you start copying some sensitive-looking files to the outer volume, the wizard tells you the maximum recommended size of space that the files should occupy, so that there is enough free space on the outer volume for the hidden volume.

Remark: After you copy some sensitive-looking files to the outer volume, the cluster bitmap of the volume will be scanned in order to determine the size of uninterrupted area of free space whose end is aligned with the end of the outer volume. This area will accommodate the hidden volume, so it limits its maximum possible size. The maximum possible size of the hidden volume will be determined and it will be verified that it is greater than the size of the system partition (which is required, because the entire content of the system partition will need to be copied to the hidden volume — see below). This ensures that no data stored on the outer volume will be overwritten by data written to the area of the hidden volume (e.g. when the system is being copied to it). The size of the hidden volume is always the same as the size of the system partition.

Then, TrueCrypt will create the hidden operating system by copying the content of the system partition to the hidden volume. Data being copied will be encrypted on the fly with an encryption key different from the one that will be used for the decoy operating system. The process of copying the system is performed in the pre-boot environment (before Windows starts) and it may take a long time to complete; several hours or even several days (depending on the size of the system partition and on the performance of the computer). You will be able to interrupt the process, shut down your computer, start the operating system and then resume the process. However, if you interrupt it, the entire process of copying the system will have to start from the beginning (because the content of the system partition must not change during cloning). The hidden operating system will initially be a clone of the operating system under which you started the wizard.

Windows creates (typically, without your knowledge or consent) various log files, temporary files, etc., on the system partition. It also saves the content of RAM to hibernation and paging files located on the system partition. Therefore, if an adversary analyzed files stored on the partition where the original system (of which the hidden system is a clone) resides, he might find out, for example, that you used the TrueCrypt wizard in the hidden-system-creation mode (which might indicate the existence of a hidden operating system on your computer). To prevent such issues, TrueCrypt will securely erase the entire content of the partition where the original system resides after the hidden system has been created. Afterwards, in order to achieve plausible deniability, TrueCrypt will prompt you to install a new system on the partition and encrypt it using TrueCrypt. Thus you will create the decoy system and the whole process of creation of the hidden operating system will be completed.

Plausible Deniability and Data Leak Protection

For security reasons, when a hidden operating system is running, TrueCrypt ensures that all local unencrypted filesystems and non-hidden TrueCrypt volumes are read-only (i.e. no files can be written to such filesystems or TrueCrypt volumes).[*] Data is allowed to be written to any filesystem that resides within a hidden TrueCrypt volume (provided that the hidden volume is not located in a container stored on an unencrypted filesystem or on any other read-only filesystem).

There are three main reasons why such countermeasures have been implemented:

1. It enables the creation of a secure platform for mounting of hidden TrueCrypt volumes. Note that we officially recommend that hidden volumes are mounted only when a hidden operating system is running. For more information, see the subsection *Security Precautions Pertaining to Hidden Volumes*.

2. In some cases, it is possible to determine that, at a certain time, a particular filesystem was not mounted under (or that a particular file on the filesystem was not saved or accessed from within) a particular instance of an operating system (e.g. by analyzing and comparing filesystem journals, file timestamps, application logs, error logs, etc). This might indicate that a hidden operating system is installed on the computer. The countermeasures prevent these issues.

3. It protects the integrity of a filesystem that is mounted under both the decoy system and the hidden system when one or both of the systems are/is hibernated.

If you need to securely transfer files from the decoy system to the hidden system, follow these steps:

1. Start the decoy system.
2. Save the files to an unencrypted volume or to an outer/normal TrueCrypt volume.
3. Start the hidden system
4. If you saved the files to a TrueCrypt volume, mount it (it will be automatically mounted as read-only).
5. Copy the files to the hidden system partition or to another hidden volume.

Possible Explanations for Existence of Two TrueCrypt Partitions on Single Drive

An adversary might ask why you created two TrueCrypt-encrypted partitions on a single drive (a system partition and a non-system partition) rather than encrypting the entire disk with a single encryption key. There are many possible reasons to do that. However, if you do not know any (other than creating the hidden operating system), you can provide, for example, one of the following explanations:

- If there are more than two partitions on a system drive and you want to encrypt only two of them (the system partition and the one behind it) and to leave the other partitions unencrypted (for example, to achieve the best possible performance when reading and writing data, which is not sensitive, to such unencrypted partitions), the only way to do that is to encrypt both partitions separately (note that, with a single encryption key, TrueCrypt could encrypt the entire system drive and *all* partitions on it, but it cannot encrypt only two of them — only one or all of the partitions can be encrypted with a single key). As a result, there will be two adjacent TrueCrypt partitions on the system drive (the first will be a system

[*] This does not apply to filesystems on CD/DVD-like media and on custom, atypical, or non-standard devices/media.

partition, the second will be a non-system one), each encrypted with a different key (which is also the case when you create a hidden operating system, and therefore it can be explained this way).

If you do not know any good reason why there should be more than one partition on a system drive at all:

It is generally recommended to separate non-system files (documents) from system files. One of the easiest and most reliable ways to do that is to create two partitions on the system drive; one for the operating system and the other for documents (non-system files). The reasons why this practice is recommended include:

- If the filesystem on one of the partitions is damaged, files on the partition may get corrupted or lost, whereas files on the other partition are not affected.
- It is easier to reinstall the system without losing your documents (reinstallation of an operating system involves formatting the system partition, after which all files stored on it are lost). If the system is damaged, full reinstallation is often the only option.

- A cascade encryption algorithm (e.g. AES-Twofish-Serpent) can be up to four times slower than a non-cascade one (e.g. AES). However, a cascade encryption algorithm may be more secure than a non-cascade one (for example, the probability that three distinct encryption algorithms will be broken, e.g. due to advances in cryptanalysis, is significantly lower than the probability that only one of them will be broken). Therefore, if you encrypt the outer volume with a cascade encryption algorithm and the decoy system with a non-cascade encryption algorithm, you can answer that you wanted the best performance (and adequate security) for the system partition, and the highest possible security (but worse performance) for the non-system partition (i.e. the outer volume), where you store the most sensitive data, which you do not need to access very often (unlike the operating system, which you use very often, and therefore you need it to have the best possible performance). On the system partition, you store data that is less sensitive (but which you need to access very often) than data you store on the non-system partition (i.e. on the outer volume).

- Provided that you encrypt the outer volume with a cascade encryption algorithm (e.g. AES-Twofish-Serpent) and the decoy system with a non-cascade encryption algorithm (e.g. AES), you can also answer that you wanted to prevent the problems about which TrueCrypt warns when the user attempts to choose a cascade encryption algorithm for system encryption (see below for a list of the problems). Therefore, to prevent those problems, you decided to encrypt the system partition with a non-cascade encryption algorithm. However, you still wanted to use a cascade encryption algorithm (because it is more secure than a non-cascade encryption algorithm) for the most sensitive data, so you decided to create a second partition, which those problems do *not* affect (because it is non-system) and to encrypt it with a cascade encryption algorithm. On the system partition, you store data that is less sensitive than data you store on the non-system partition (i.e. on the outer volume).

  Note: When the user attempts to encrypt the system partition with a cascade encryption algorithm, TrueCrypt warns him or her that it can cause the following problems (and implicitly recommends to choose a non-cascade encryption algorithm instead):
  - For cascade encryption algorithms, the TrueCrypt Boot Loader is larger than normal and, therefore, there is not enough space in the first drive track for a backup of the TrueCrypt Boot Loader. Hence, *whenever* it gets damaged (which often happens, for example, during inappropriately designed anti-piracy activation procedures of certain programs), the user must use the TrueCrypt Rescue Disk to repair the TrueCrypt Boot Loader or to boot.
  - Due to increased memory requirements, on some computers, it is impossible to encrypt the system partition/drive.

    - On some computers, resuming from hibernation takes longer.

- In contrast to a password for a non-system TrueCrypt volume, a pre-boot authentication password needs to be typed each time the computer is turned on or restarted. Therefore, if the pre-boot authentication password is long (which is required for security purposes), it may be very tiresome to type it so frequently. Hence, you can answer that it was more convenient for you to use a short (and therefore weaker) password for the system partition (i.e. the decoy system) and that it is more convenient for you to store the most sensitive data (which you do not need to access as often) in the non-system TrueCrypt partition (i.e. in the outer volume) for which you chose a very long password.

  As the password for the system partition is not very strong (because it is short), you do not intentionally store sensitive data on the system partition. However, you still prefer the system partition to be encrypted, because potentially sensitive or mildly sensitive data is stored on it as a result of your everyday use of the computer (for example, passwords to online forums you visit, which can be automatically remembered by your browser, browsing history, applications you run, etc.)

- When an attacker gets hold of your computer when a TrueCrypt volume is mounted (for example, when you use a laptop outside), he can, in most cases, read any data stored on the volume (data is decrypted on the fly as he reads it). Therefore, it may be wise to limit the time the volume is mounted to a minimum. Obviously, this may be impossible or difficult if the sensitive data is stored on an encrypted system partition or on an entirely encrypted system drive (because you would also have to limit the time you work with the computer to a minimum). Hence, you can answer that you created a separate partition (encrypted with a different key than your system partition) for your most sensitive data and that you mount it only when necessary and dismount it as soon as possible (so as to limit the time the volume is mounted to a minimum). On the system partition, you store data that is less sensitive (but which you need to access often) than data you store on the non-system partition (i.e. on the outer volume).

Safety and Security Precautions Pertaining to Hidden Operating Systems

As a hidden operating system resides in a hidden TrueCrypt volume, a user of a hidden operating system should follow all of the security precautions that apply to normal hidden TrueCrypt volumes. These precautions, as well as additional precautions pertaining specifically to hidden operating systems, are listed in the subsection *Security Precautions Pertaining to Hidden Volumes*.

WARNING: If you do not protect the hidden volume (for information on how to do so, refer to the section *Protection of Hidden Volumes Against Damage*), do *not* write to the outer volume (note that the decoy operating system is *not* installed in the outer volume). Otherwise, you may overwrite and damage the hidden volume (and the hidden operating system within it)!

If all the instructions in the wizard have been followed and if the security precautions mentioned in the subsection *Security Precautions Pertaining to Hidden Volumes* are followed, it will be impossible to prove that the hidden volume and hidden operating system exist, even when the outer volume is mounted or when the decoy operating system is decrypted or started.

System Encryption

TrueCrypt can on-the-fly encrypt a system partition or entire system drive, i.e. a partition or drive where Windows is installed and from which it boots.

System encryption provides the highest level of security and privacy, because all files, including any temporary files that Windows and applications create on the system partition (typically, without your knowledge or consent), hibernation files, swap files, etc., are always permanently encrypted (even when power supply is suddenly interrupted). Windows also records large amounts of potentially sensitive data, such as the names and locations of files you open, applications you run, etc. All such log files and registry entries are always permanently encrypted as well.

System encryption involves pre-boot authentication, which means that anyone who wants to gain access and use the encrypted system, read and write files stored on the system drive, etc., will need to enter the correct password each time before Windows boots (starts). Pre-boot authentication is handled by the TrueCrypt Boot Loader, which resides in the first track of the boot drive and on the TrueCrypt Rescue Disk (see below).

Note that TrueCrypt can encrypt an existing unencrypted system partition/drive in-place while the operating system is running (while the system is being encrypted, you can use your computer as usual without any restrictions). Likewise, a TrueCrypt-encrypted system partition/drive can be decrypted in-place while the operating system is running. You can interrupt the process of encryption or decryption anytime, leave the partition/drive partially unencrypted, restart or shut down the computer, and then resume the process, which will continue from the point it was stopped.

The mode of operation used for system encryption is XTS (see the section *Modes of Operation*). For further technical details of system encryption, see the section *Encryption Scheme* in the chapter *Technical Details*.

To encrypt a system partition or entire system drive, select *System* > *Encrypt System Partition/Drive* and then follow the instructions in the wizard. To decrypt a system partition/drive, select *System* > *Permanently Decrypt System Partition/Drive*.

Operating Systems Supported for System Encryption

TrueCrypt can currently encrypt the following operating systems:

- Windows Vista
- Windows Vista x64 (64-bit) Edition
- Windows XP
- Windows XP x64 (64-bit) Edition
- Windows Server 2008
- Windows Server 2008 x64 (64-bit)
- Windows Server 2003
- Windows Server 2003 x64 (64-bit)

TrueCrypt Rescue Disk

During the process of preparing the encryption of a system partition/drive, TrueCrypt requires that you create a so-called TrueCrypt Rescue Disk (CD/DVD), which serves the following purposes:

- If the TrueCrypt Boot Loader screen does not appear after you start your computer (or if your Windows does not boot), the **TrueCrypt Boot Loader may be damaged**. The TrueCrypt Rescue Disk allows you restore it and thus to regain access to your encrypted system and data (however, note that you will still have to enter the correct password then). In the Rescue Disk screen, select *Repair Options* > *Restore TrueCrypt Boot Loader*. Then press 'Y' to confirm the action, remove the Rescue Disk from your CD/DVD drive and restart your computer.

- If you repeatedly enter the correct password but TrueCrypt says that the password is incorrect, it is possible that the **master key or other critical data are damaged**. The TrueCrypt Rescue Disk allows you to restore them and thus to regain access to your encrypted system and data (however, note that you will still have to enter the correct password then). In the Rescue Disk screen, select *Repair Options* > *Restore key data*. Then enter your password, press 'Y' to confirm the action, remove the Rescue Disk from your CD/DVD drive, and restart your computer.

  Note: This feature cannot be used to restore the header of a hidden volume within which a hidden operating system resides (see the section *Hidden Operating System*). To restore such a volume header, click *Select Device*, select the partition behind the boot partition, click *OK*, select *Tools -> Restore Volume Header* and then follow the instructions.

  WARNING: By restoring key data using a TrueCrypt Rescue Disk, you also restore the password that was valid when the TrueCrypt Rescue Disk was created. Therefore, whenever you change the password, you should destroy your TrueCrypt Rescue Disk and create a new one (select *System -> Create Rescue Disk*). Otherwise, if an attacker knows your old password (for example, captured by a keystroke logger) and if he then finds your old TrueCrypt Rescue Disk, he could use it to restore the key data (the master key encrypted with the old password) and thus decrypt your system partition/drive.

- If the **TrueCrypt Boot Loader is damaged or infected with malware**, you can avoid running it by booting directly from the TrueCrypt Rescue Disk. Insert your Rescue Disk into your CD/DVD drive and then enter your password in the Rescue Disk screen.

- If **Windows is damaged and cannot start**, the TrueCrypt Rescue Disk allows you to permanently decrypt the partition/drive before Windows starts. In the Rescue Disk screen, select *Repair Options* > *Permanently decrypt system partition/drive*. Enter the correct password and wait until decryption is complete. Then you can e.g. boot your MS Windows setup CD/DVD to repair your Windows. Note that this feature cannot be used to decrypt a hidden volume within which a hidden operating system resides (see the section *Hidden Operating System*).

  Note: Alternatively, if Windows is damaged (cannot start) and you need to repair it (or access files on it), you can avoid decrypting the system partition/drive by following these steps: If you have multiple operating systems installed on your computer, boot the one that does not require pre-boot authentication. If you do not have multiple operating systems installed on your computer, you can boot a WinPE or BartPE CD/DVD or you can connect your system drive as a secondary or external drive to another computer and then boot the operating system installed on the computer. After you boot a system, run TrueCrypt, click

Select Device, select the affected system partition, click *OK* , select *System* > *Mount Without Pre-Boot Authentication*, enter your pre-boot-authentication password and click *OK*. The partition will be mounted as a regular TrueCrypt volume (data will be on-the-fly decrypted/encrypted in RAM on access, as usual).

- Your TrueCrypt Rescue Disk contains a **backup of the original content of the first drive track** (made before the TrueCrypt Boot Loader was written to it) and allows you to restore it if necessary. The first track typically contains a system loader or boot manager. In the Rescue Disk screen, select *Repair Options* > *Restore original system loader*.

*Note that even if you lose your TrueCrypt Rescue Disk and an attacker finds it, he or she will **not** be able to decrypt the system partition or drive without the correct password.*

To boot a TrueCrypt Rescue Disk, insert it into your CD/DVD drive and restart your computer. If the TrueCrypt Rescue Disk screen does not appear (or if you do not see the 'Repair Options' item in the 'Keyboard Controls' section of the screen), it is possible that your BIOS is configured to attempt to boot from hard drives before CD/DVD drives. If that is the case, restart your computer, press F2 or Delete (as soon as you see a BIOS start-up screen), and wait until a BIOS configuration screen appears. If no BIOS configuration screen appears, restart (reset) the computer again and start pressing F2 or Delete repeatedly as soon as you restart (reset) the computer. When a BIOS configuration screen appears, configure your BIOS to boot from the CD/DVD drive first (for information on how to do so, please refer to the documentation for your BIOS/motherboard or contact your computer vendor's technical support team for assistance). Then restart your computer. The TrueCrypt Rescue Disk screen should appear now. Note: In the TrueCrypt Rescue Disk screen, you can select 'Repair Options' by pressing F8 on your keyboard.

If your Rescue Disk is damaged, you can create a new one by selecting *System* > *Create Rescue Disk*. To find out whether your TrueCrypt Rescue Disk is damaged, insert it into your CD/DVD drive and select *System* > *Verify Rescue Disk*.

Hidden Operating System

It may happen that you are forced by somebody to decrypt the operating system. There are many situations where you cannot refuse to do so (for example, due to extortion). TrueCrypt allows you to create a hidden operating system whose existence will be impossible to prove (provided that certain guidelines are followed). Thus, you will not have to decrypt or reveal the password for the hidden operating system. For more information, see the section *Hidden Operating System* in the chapter *Plausible Deniability*.

TrueCrypt Volume

There are two types of TrueCrypt volumes:

- File-hosted (container)
- Partition/device-hosted (non-system)

Note: In addition to creating the above types of virtual volumes, TrueCrypt can encrypt a physical partition/drive where Windows is installed (for more information, see the chapter *System Encryption*).

A TrueCrypt file-hosted volume is a normal file, which can reside on any type of storage device. It contains (hosts) a completely independent encrypted virtual disk device.

A TrueCrypt partition is a hard disk partition encrypted using TrueCrypt. You can also encrypt entire hard disks, USB hard disks, floppy disks, USB memory sticks, and other types of storage devices.

Creating a New TrueCrypt Volume

To create a new TrueCrypt file-hosted volume or to encrypt a partition/device (requires administrator privileges), click on 'Create Volume' in the main program window. TrueCrypt Volume Creation Wizard should appear. As soon as the Wizard appears, it starts collecting data that will be used in generating the master key, secondary key (XTS mode), and salt, for the new volume. The collected data, which should be as random as possible, include your mouse movements, key presses, and other values obtained from the system (for more information, please see the section *Random Number Generator*). The Wizard provides help and information necessary to successfully create a new TrueCrypt volume. However, several items deserve further explanation:

Hash Algorithm

Allows you to select which hash algorithm TrueCrypt will use. The selected hash algorithm is used by the random number generator (as a pseudorandom mixing function), which generates the master key, secondary key (XTS mode), and salt (for more information, please see the section *Random Number Generator*). It is also used in deriving the new volume header key and secondary header key (see the section *Header Key Derivation, Salt, and Iteration Count*).

For information about the implemented hash algorithms, see the chapter *Hash Algorithms.*

Note that the output of a hash function is *never* used directly as an encryption key. For more information, please refer to the chapter *Technical Details*.

Encryption Algorithm

This allows you to select the encryption algorithm with which your new volume will be encrypted. Note that the encryption algorithm cannot be changed after the volume is created. For more information, please see the chapter *Encryption Algorithms*.

Quick Format

If unchecked, each sector of the new volume will be formatted. This means that the new volume will be *entirely* filled with random data. Quick format is much faster but may be less secure because until the whole volume has been filled with files, it may be possible to tell how much data it contains (if the space was not filled with random data beforehand). If you are not sure whether to enable or disable Quick Format, we recommend that you leave this option unchecked. Note that Quick Format can only be enabled when encrypting partitions/devices.
Important: When encrypting a partition/device within which you intend to create a hidden volume afterwards, leave this option unchecked.

Dynamic

Dynamic TrueCrypt container is a pre-allocated NTFS sparse file whose physical size (actual disk space used) grows as new data is added to it. Note that the physical size of the container (actual disk space that the container uses) will not decrease when files are deleted on the TrueCrypt volume. The physical size of the container can only *increase* up to the maximum value that is specified by the user during the volume creation process. After the maximum specified size is reached, the physical size of the container will remain constant.

Note that sparse files can only be created in the NTFS file system. If you are creating a container in the FAT file system, the option *Dynamic* will be disabled ("grayed out").

Note that the size of a dynamic (sparse-file-hosted) TrueCrypt volume reported by Windows and by TrueCrypt will always be equal to its maximum size (which you specify when creating the volume). To find out current physical size of the container (actual disk space it uses), right-click the container file (in a Windows Explorer window, not in TrueCrypt), then select *Properties* and see the *Size on disk* value.

WARNING: Performance of dynamic (sparse-file-hosted) TrueCrypt volumes is significantly worse than performance of regular volumes. Dynamic (sparse-file-hosted) TrueCrypt volumes are also less secure, because it is possible to tell which volume sectors are unused. Furthermore, if data is written to a dynamic volume when there is not enough free space in its host file system, the encrypted file system may get corrupted.

Cluster Size

Cluster is an allocation unit. For example, one cluster is allocated on a FAT file system for a one-byte file. When the file grows beyond the cluster boundary, another cluster is allocated. Theoretically, this means that the bigger the cluster size, the more disk space is wasted; however, the better the performance. If you do not know which value to use, use the default.

TrueCrypt Volumes on CDs and DVDs

If you want a TrueCrypt volume to be stored on a CD or a DVD, first create a file-hosted TrueCrypt container on a hard drive and then burn it onto a CD/DVD using any CD/DVD burning software (or, under Windows XP/Vista, using the CD burning tool provided with the operating system). Remember that if you need to mount a TrueCrypt volume that is stored on a read-only medium (such as a CD/DVD) under Windows 2000, you must format the TrueCrypt volume as FAT. The

reason is that Windows 2000 cannot mount NTFS file system on read-only media (Windows XP/Vista can).

Hardware/Software RAID, Windows Dynamic Volumes

TrueCrypt supports hardware/software RAID as well as Windows dynamic volumes.

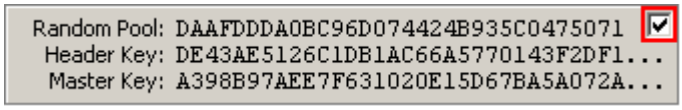
Windows Vista or later: Dynamic volumes are displayed in the 'Select Device' dialog window as `\Device\HarddiskVolume`*N*.

Windows XP/2000/2003: If you intend to format a Windows dynamic volume as a TrueCrypt volume, keep in mind that after you create the Windows dynamic volume (using the Windows Disk Management tool), you must restart the operating system in order for the volume to be available/displayed in the 'Select Device' dialog window of the TrueCrypt Volume Creation Wizard. Also note that, in the 'Select Device' dialog window, a Windows dynamic volume is *not* displayed as a single device (item). Instead, *all* volumes that the Windows dynamic volume consists of are displayed and you can select *any* of them in order to format the *entire* Windows dynamic volume.

Additional Notes on Volume Creation

After you click the 'Format' button in the Volume Creation Wizard window (the last step), there will be a short delay while your system is being polled for additional random data. Afterwards, the master key, header key, secondary key (XTS mode), and salt, for the new volume will be generated, and the master key and header key contents will be displayed.

For extra security, the randomness pool, master key, and header key contents can be prevented from being displayed by unchecking the checkbox in the upper right corner of the corresponding field:



Note that only the first 128 bits of the pool/keys are displayed (not the entire contents).

You can create FAT (whether it will be FAT12, FAT16, or FAT32, is automatically determined from the number of clusters) or NTFS volumes (however, NTFS volumes can only be created by users with administrator privileges). Mounted TrueCrypt volumes can be reformatted as FAT12, FAT16, FAT32, or NTFS anytime. They behave as standard disk devices so you can right-click the drive letter of the mounted TrueCrypt volume (for example in the '*Computer*' or '*My Computer*' list) and select 'Format'.

For more information about creating TrueCrypt volumes, see also the section *Hidden Volume*.

Main Program Window

Select File

Allows you to select a file-hosted TrueCrypt volume. After you select it, you can perform various operations on it (e.g., mount it by clicking ‘Mount’). It is also possible to select a volume by dragging its icon to the ‘TrueCrypt.exe’ icon (TrueCrypt will be automatically launched then) or to the main program window.

Select Device

Allows you to select a TrueCrypt partition or a storage device (such as floppy disk or USB memory stick). After it is selected, you can perform various operations with it (e.g., mount it by clicking ‘Mount’).

Note: There is a more comfortable way of mounting TrueCrypt partitions/devices – see the section *Auto-Mount Devices* for more information.

Mount

After you click ‘Mount’, TrueCrypt will try to mount the selected volume using cached passwords (if there are any) and if none of them works, it prompts you for a password. If you enter the correct password (and/or provide correct keyfiles), the volume will be mounted.
Important: Note that when you exit the TrueCrypt application, the TrueCrypt driver continues working and no TrueCrypt volume is dismounted.

Auto-Mount Devices

This function allows you to mount TrueCrypt partitions/devices without having to select them manually (by clicking ‘Select Device’). TrueCrypt scans headers of all available partitions/devices on your system one by one and tries to mount each of them as a TrueCrypt volume. Note that TrueCrypt partition/device cannot be identified, nor the cipher it has been encrypted with. Therefore, the program cannot directly “find” TrueCrypt partitions. Instead, it has to try mounting each (even unencrypted) partition/device using all encryption algorithms and all cached passwords (if there are any). Therefore, be prepared that this process may take a long time on slow computers.

If the password you enter is wrong, mounting is attempted using cached passwords (if there are any). If you enter an empty password and if *Use keyfiles* is unchecked, only the cached passwords will be used when attempting to auto-mount partitions/devices. If you do not need to set mount options, you can bypass the password prompt by holding down the *Shift* key when clicking *Auto-Mount Devices* (only cached passwords will be used, if there are any).

Drive letters will be assigned starting from the one that is selected in the drive list in the main window.

Dismount

To dismount a TrueCrypt volume means to close it and make it impossible to read/write from/to the volume.

Dismount All

To dismount a TrueCrypt volume means to close it and make it impossible to read/write from/to the volume. This function dismount all currently mounted TrueCrypt volumes.

Wipe Cache

Clears all passwords (which may also contain processed keyfile contents) cached in driver memory. When there are no passwords in the cache, this button is disabled. For information on password cache, see the section *Cache Password in Driver Memory*.

Never Save History

If this option disabled, the file names and/or paths of the last twenty files/devices that were attempted to be mounted as TrueCrypt volumes will be saved in the History file (whose content can be displayed by clicking on the Volume combo-box in the main window). When this option is enabled, TrueCrypt clears the registry entries created by the Windows file selector for TrueCrypt, and sets the “current directory” to the user’s home directory (in traveler mode, to the directory from which TrueCrypt was launched) whenever a container or keyfile is selected via the Windows file selector. Therefore, the Windows file selector will not remember the path of the last mounted container (or the last selected keyfile). Furthermore, if this option is enabled, the volume path input field in the main TrueCrypt window is cleared whenever you hide TrueCrypt.

Note: You can clear the volume history by selecting *Tools -> Clear Volume History*.

Exit

Terminates the TrueCrypt application. The driver continues working and no TrueCrypt volumes are dismounted. When running in 'traveler' mode, the TrueCrypt driver is unloaded when it is no longer needed (e.g., when all instances of the main application and/or of the Volume Creation Wizard are closed and no TrueCrypt volumes are mounted). However, if you force dismount on a TrueCrypt volume when TrueCrypt runs in ‘traveler’ mode, the TrueCrypt driver will *not* be unloaded when you exit TrueCrypt (it will be unloaded only when you shut down or restart the system). This prevents various problems caused by a bug in Windows (for instance, it would be impossible to start TrueCrypt again as long as there are applications using the dismounted volume).

Volume Tools

Change Volume Password

See the section *Volumes -> Change Volume Password*.

Set Header Key Derivation Algorithm

See the section *Volumes -> Set Header Key Derivation Algorithm*.

Backup Volume Header

See the section *Tools -> Backup Volume Header*.

Restore Volume Header

See the section *Tools -> Restore Volume Header*.

Program Menu

Note: To save space, only the menu items that are not self-explanatory are described in this documentation.

Volumes -> Auto-Mount All Device-Hosted Volumes

See the section *Auto-Mount Devices.*

Volumes -> Save Currently Mounted Volumes as Favorite

This function is useful if you often work with more than one TrueCrypt volume at a time and you need each of them to be always mounted to a particular drive letter.

A list of all currently mounted volumes (and the drive letters they are mounted as) is saved to a file called *Favorite Volumes.xml* in the folder where application data are saved on your system (for example, in *C:\Documents and Settings\YourUserName\Application Data\TrueCrypt*). In traveler mode, the file is saved to the folder from which you run the file *TrueCrypt.exe* (in which *TrueCrypt.exe* resides).

Note that when you use this function, all dismounted volumes that were previously saved as favorite will be deleted from the list of favorite volumes.

To mount volumes saved as “Favorite”, select *Volumes -> Mount Favorite Volumes*

To delete the list of favorite volumes, dismount all TrueCrypt volumes, and select *Volumes -> Save Currently Mounted Volumes as Favorite*.

Volumes -> Mount Favorite Volumes

This function mounts volumes you previously saved as “Favorite”. For more information, see the section *Volumes -> Save Currently Mounted Volumes as Favorite* above.

Volumes -> Set Header Key Derivation Algorithm

This function allows you to re-encrypt a volume header with a header key derived using a different PRF function (for example, instead of HMAC-RIPEMD-160 you could use HMAC-Whirlpool). Note that the volume header contains the master encryption key with which the volume is encrypted. Therefore, the data stored on the volume will *not* be lost after you use this function. For more information, see the section *Header Key Derivation, Salt, and Iteration Count*.

Note: When TrueCrypt re-encrypts a volume header, the original volume header is first overwritten 200 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunneling microscopy [17] to recover the overwritten header (however, see also the chapter *Security Precautions*).

Volumes -> Change Volume Password

Allows changing the password of the currently selected TrueCrypt volume (no matter whether the volume is hidden or standard). Only the header key and the secondary header key (XTS mode) are changed – the master key remains unchanged. This function re-encrypts the volume header using a header encryption key derived from a new password. Note that the volume header contains the master encryption key with which the volume is encrypted. Therefore, the data stored on the volume will *not* be lost after you use this function (password change will only take a few seconds).

To change a TrueCrypt volume password, click on *Select File* or *Select Device*, then select the volume, and from the *Volumes* menu select *Change Volume Password*.

Note: For information on how to change a password used for pre-boot authentication, please see the section *System -> Change Password*.

See also the chapter *Security Precautions*.

PKCS-5 PRF

In this field you can select the algorithm that will be used in deriving new volume header keys (for more information, see the section *Header Key Derivation, Salt, and Iteration Count*) and in generating the new salt (for more information, see the section *Random Number Generator*).

Note: When TrueCrypt re-encrypts a volume header, the original volume header is first overwritten 200 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunneling microscopy [17] to recover the overwritten header (however, see also the chapter *Security Precautions*).

System -> Change Password

Changes the password used for pre-boot authentication (see the chapter *System Encryption*).

WARNING: Your TrueCrypt Rescue Disk allows you to restore key data if it is damaged. By doing so, you also restore the password that was valid when the TrueCrypt Rescue Disk was created. Therefore, whenever you change the password, you should destroy your TrueCrypt Rescue Disk and create a new one (select *System -> Create Rescue Disk*). Otherwise, an attacker could decrypt your system partition/drive using the old password (if he finds the old TrueCrypt Rescue Disk and uses it to restore the key data). See also the chapter *Security Precautions*.

For more information on changing a password, please see the section *Volumes -> Change Volume Password* above.

System -> Mount Without Pre-Boot Authentication

Check this option, if you need to mount a partition that is within the key scope of system encryption without pre-boot authentication. For example, if you need to mount a partition located on the encrypted system drive of another operating system that is not running. This can be useful e.g. when you need to back up or repair an operating system encrypted by TrueCrypt (from within another operating system).

Note 1: If you need to mount multiple partitions at once, click *'Auto-Mount Devices*', then click '*Mount Options*' and enable the option '*Mount partition using system encryption without pre-boot authentication*'.

Please note you cannot use this function to mount extended (logical) partitions that are located on an entirely encrypted system drive.

Tools -> Clear Volume History

Clears the list containing the file names (if file-hosted) and paths of the last twenty successfully mounted volumes.

Tools -> Traveler Disk Setup

See the chapter *Traveler Mode.*

Tools -> Keyfile Generator

See the section *Keyfiles -> Generate Random Keyfile.*

Tools -> Backup Volume Header

Tools -> Restore Volume Header

If the header of a TrueCrypt volume is damaged, the volume is, in most cases, impossible to mount. Therefore, each volume created by TrueCrypt 6.0 or later contains an embedded backup header, located at the end of the volume. For extra safety, you can also create external volume header backup files. To do so, click *Select Device* or *Select File*, select the volume, select *Tools -> Backup Volume Header*, and then follow the instructions.

Note: A backup header (embedded or external) is *not* a copy of the original volume header because it is encrypted with a different header key derived using a different salt (see the section *Header Key Derivation, Salt, and Iteration Count*). When the volume password and/or keyfiles are changed, or when the header is restored from the embedded (or an external) header backup, both the volume header and the backup header (embedded in the volume) are re-encrypted with header keys derived using newly generated salts (the salt for the volume header is different from the salt for the backup header). Each salt is generated by the TrueCrypt random number generator (see the section *Random Number Generator*).

Both types of header backups (embedded and external) can be used to repair a damaged volume header. To do so, click *Select Device* or *Select File*, select the volume, select *Tools -> Restore Volume Header*, and then follow the instructions.

WARNING: Restoring a volume header also restores the volume password that was valid when the backup was created. Moreover, if keyfile(s) are/is necessary to mount a volume when the backup is created, the same keyfile(s) will be necessary to mount the volume again after the volume header is restored. For more information, see the section *Encryption Scheme* in the chapter *Technical Details*.

After you create a volume header backup, you might need to create a new one only when you change the volume password and/or keyfiles. Otherwise, the volume header remains unmodified so the volume header backup remains up-to-date.

Note: Apart from salt (which is a sequence of random numbers), external header backup files do not contain any unencrypted information and they cannot be decrypted without knowing the correct password and/or supplying the correct keyfile(s). For more information, see the chapter *Technical Details*.

When you create an external header backup, both the standard volume header and the area where a hidden volume header can be stored is backed up, even if there is no hidden volume within the volume (to preserve plausible deniability of hidden volumes). If there is no hidden volume within the volume, the area reserved for the hidden volume header in the backup file will be filled with random data (to preserve plausible deniability).

When *restoring* a volume header, you need to choose the type of volume whose header you wish to restore (a standard or hidden volume). Only one volume header can be restored at a time. To restore both headers, you need to use the function twice (*Tools -> Restore Volume Header*). You will need to enter the correct password (and/or to supply the correct keyfiles) that was/were valid when the volume header backup was created. The password (and/or keyfiles) will also automatically determine the type of the volume header to restore, i.e. standard or hidden (note that TrueCrypt determines the type through the process of trial and error).

Note: If the user fails to supply the correct password (and/or keyfiles) twice in a row when trying to mount a volume, TrueCrypt will automatically try to mount the volume using the embedded backup header (in addition to trying to mount it using the primary header) each subsequent time that the user attempts to mount the volume (until he or she clicks *Cancel*). If TrueCrypt fails to decrypt the primary header but it successfully decrypts the embedded backup header at the same time, the volume is mounted and the user is warned that the volume header is damaged (and informed as to how to repair it).

Note that these features can be used in a corporate environment to reset volume passwords in case a user forgets it (or when he/she loses his/her keyfile). After you create a volume, backup its header (select *Tools -> Backup Volume Header*) before you allow a non-admin user to use the volume. Note that the volume header (which is encrypted with a header key derived from a password/keyfile) contains the master key with which the volume is encrypted. Then ask the user to choose a password, and set it for him/her (*Volumes -> Change Volume Password*); or generate a user keyfile for him/her. Then you can allow the user to use the volume and to change the password/keyfiles without your assistance/permission. In case he/she forgets his/her password or loses his/her keyfile, you can "reset" the volume password/keyfiles to your original admin password/keyfiles by restoring the volume header backup (*Tools -> Restore Volume Header*).

Settings -> Preferences

Invokes the Preferences dialog window, where you can change, among others, the following options:

Wipe cached passwords on exit

If enabled, passwords (which may also contain processed keyfile contents) cached in driver memory will be cleared when TrueCrypt exits.

Cache passwords in driver memory

When checked, passwords and/or processed keyfile contents for up to last four successfully mounted TrueCrypt volumes are cached. This allows mounting volumes without having to type their passwords (and selecting keyfiles) repeatedly. TrueCrypt never saves any password to a disk (however, see the chapter *Security Precautions*). Password caching can be enabled/disabled in the Preferences (*Settings -> Preferences*) and in the password prompt window.

Open Explorer window for successfully mounted volume

If this option is checked, then after a TrueCrypt volume has been successfully mounted, an Explorer window showing the root directory of the volume (e.g., `T:\`) will be automatically opened.

Close all Explorer windows of volume being dismounted

Sometimes, dismounting a TrueCrypt volume is not possible because some files or folders located on the volume are in use or "locked". This also applies to Explorer windows displaying directories located on TrueCrypt volumes. When this option is checked, all such windows will be automatically closed before dismounting, so that the user does not have to close them manually.

TrueCrypt Background Task – Enabled

See the chapter *TrueCrypt Background Task*.

TrueCrypt Background Task – Exit when there are no mounted volumes

If this option is checked, the TrueCrypt background task automatically and silently exits as soon as there are no mounted TrueCrypt volumes. For more information, see the chapter *TrueCrypt Background Task*. Note that this option cannot be disabled when TrueCrypt runs in traveler mode.

Auto-dismount volume after no data has been read/written to it for

After no data has been written/read to/from a TrueCrypt volume for *n* minutes, the volume is automatically dismounted.

Force auto-dismount even if volume contains open files or directories

This option applies only to auto-dismount (not to regular dismount). It forces dismount (without prompting) on the volume being auto-dismounted in case it contains open files or directories (i.e., file/directories that are in use by the system or applications).

Mounting TrueCrypt Volumes

If you have not done so yet, please read the sections '*Mount*' and '*Auto-Mount Devices*' in the chapter *Main Program Window*.

Cache Password in Driver Memory

This option can be set in the password entry dialog so that it will apply only to that particular mount attempt. It can also be set as default in the Preferences. For more information, please see the section *Settings -> Preferences*, subsection *Cache passwords in driver memory*.

Mount Options

Mount options affect the parameters of the volume being mounted. The *Mount Options* dialog can be opened by clicking on the *Mount Options* button in the password entry dialog. When a correct password is cached, volumes are automatically mounted after you click *Mount*. If you need to change mount options for a volume being mounted using a cached password, hold down the *Control* (*Ctrl*) key while clicking *Mount,* or select *Mount with Options* from the *Volumes* menu.

Default mount options can be configured in the main program preferences (*Settings -> Preferences).*

Mount volume as read-only

When checked, it will not be possible to write any data to the mounted volume. Note that Windows 2000 do not allow NTFS volumes to be mounted as read-only.

Mount volume as removable medium

Check this option, for example, if you need to prevent Windows from automatically creating the '*Recycled*' and/or '*System Volume Information*' folders on the volume (these folders are used by the Recycle Bin and System Restore facilities).

Use backup header embedded in volume if available

All volumes created by TrueCrypt 6.0 or later contain an embedded backup header (located at the end of the volume). If you check this option, TrueCrypt will attempt to mount the volume using the embedded backup header. Note that if the volume header is damaged, you do not have to use this option. Instead, you can repair the header by selecting *Tools > Restore Volume Header*.

Mount partition using system encryption without pre-boot authentication

Check this option, if you need to mount a partition that is within the key scope of system encryption without pre-boot authentication. For example, if you need to mount a partition located on the encrypted system drive of another operating system that is not running. This can be useful e.g. when you need to back up or repair an operating system encrypted by TrueCrypt (from within another operating system). Note that this option can be enabled also when using the '*Auto-Mount Devices*' or '*Auto-Mount All Device-Hosted Volumes*' functions.

Hidden Volume Protection

Please see the section *Protection of Hidden Volumes Against Damage*.

Hot Keys

To set system-wide TrueCrypt hot keys, click *Settings -> Hot Keys*. Note that hot keys work only when TrueCrypt or the TrueCrypt Background Task is running.

Keyfiles

Keyfile is a file whose content is combined with a password (for information on the method used to combine a keyfile with password, see the chapter *Technical Details*, section *Keyfiles*). Until the correct keyfile is provided, no volume that uses the keyfile can be mounted.

You do not have to use keyfiles. However, using keyfiles has some advantages:

- Provides protection against keystroke loggers (even if an adversary captures your password using a keystroke logger, he will not be able to mount the volume without your keyfile).
- May improve protection against brute force attacks (significant particularly if the volume password is weak).
- Allows managing multi-user *shared* access (all keyfile holders must present their keyfiles before a volume can be mounted).

Any kind of file (for example, .txt, .exe, mp3[*], .avi) may be used as a TrueCrypt keyfile (however, we recommend that you prefer compressed files, such as .mp3, .jpg, .zip, etc). Note that TrueCrypt never modifies the keyfile contents. Therefore, it is possible to use, for example, five files in your large mp3 collection as TrueCrypt keyfiles (and inspection of the files will not reveal that they are used as keyfiles).

You can select more than one keyfile; the order does not matter. You can also let TrueCrypt generate a file with random content and use it as a keyfile. To do so, select *Keyfiles -> Generate Random Keyfile*.

IMPORTANT: To make brute force attacks on a keyfile infeasible, the size of the keyfile should be at least 30 bytes. If a volume uses multiple keyfiles, then at least one of the keyfiles should be 30 bytes in size or larger. Note that the 30-byte limit assumes a large amount of entropy in the keyfile. If the first 1024 kilobytes of a file contain only a small amount of entropy, it should not be used as a keyfile (regardless of the file size). If you are not sure what entropy means, we recommend that you let TrueCrypt generate a file with random content and that you use it as a keyfile (select *Keyfiles -> Generate Random Keyfile*).

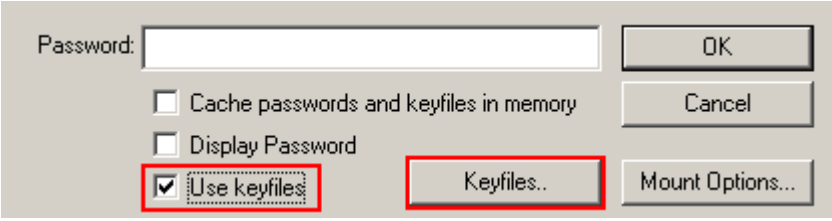
WARNING: If you lose a keyfile or if any bit of its first 1024 kilobytes changes, it will be impossible to mount volumes that use the keyfile!

WARNING: If password caching is enabled, the password cache also contains the processed contents of keyfiles used to successfully mount a volume. Then it is possible to remount the volume even if the keyfile is not available/accessible. *To prevent this, click '*Wipe Cache*' or disable password caching (for more information, please see the section* Settings -> Preferences, *subsection* Cache passwords in driver memory).

[*] However, if you use an MP3 file as a keyfile, you must ensure that no program modifies the ID3 tags (e.g. song title, name of artist, etc.) within the MP3 file. Otherwise, it will be impossible to mount volumes that use the keyfile.

Keyfiles Dialog Window

If you want to use keyfiles (i.e. “apply” them) when creating or mounting volumes, or changing passwords, look for the ‘*Use keyfiles*’ option and the *Keyfiles* button below a password input field.



These control elements appear in various dialog windows and always have the same functions. Check the *Use keyfiles* option and click *Keyfiles.* The keyfile dialog window should appear where you can specify keyfiles (to do so, click *Add Files* or *Add Token Files*) *or* keyfile search paths (click *Add Path*).

Security Tokens and Smart Cards

TrueCrypt can directly use keyfiles stored on a security token or smart card that complies with the PKCS #11 standard [23] and that allows the user to store a file (data object) on the token/card. To use such files as TrueCrypt keyfiles, click *Add Token Files* (in the keyfile dialog window).

Access to a keyfile stored on a security token or smart card is typically protected by PIN codes, which can be entered either using a hardware PIN pad or via the TrueCrypt GUI. It can also be protected by other means, such as fingerprint readers.

In order to allow TrueCrypt to access a security token or smart card, you need to install a PKCS #11 software library for the token or smart card first. Such a library may be supplied with the device or it may be available for download from the website of the vendor or other third parties.

If your security token or smart card does not contain any file (data object) that you could use as a TrueCrypt keyfile, you can use TrueCrypt to import any file to the token or smart card (if it is supported by the device). To do so, follow these steps:

1. In the keyfile dialog window, click *Add Token Files*.
2. If the token or smart card is protected by a PIN, password, or other means (such as a fingerprint reader), authenticate yourself (for example, by entering the PIN using a hardware PIN pad).
3. The ‘Security Token Keyfile’ dialog window should appear. In it, click *Import Keyfile to Token* and then select the file you want to import to the token or smart card.

Note that you can import for example 512-bit keyfiles with random content generated by TrueCrypt (see *Keyfiles -> Generate Random Keyfile* below).

Keyfile Search Path

By adding a folder in the keyfile dialog window (click *Add Path*), you specify a *keyfile search path*. All files found in the keyfile search path* will be used as keyfiles.

Important: Note that folders (and files they contain) found in keyfile search paths are ignored.

Keyfile search paths are especially useful if you, for example, store keyfiles on a USB memory stick that you carry with you. You can set the drive letter of the USB memory stick as a default keyfile search path. To do so, select *Keyfiles -> Set Default Keyfiles/Paths*. Then click *Add Path*, browse to the drive letter assigned to the USB memory stick, and click *OK*. Now each time you mount a volume (and if the option *Use keyfiles* is checked in the password dialog window), TrueCrypt will scan the path and use all files that it finds on the USB memory stick as keyfiles.

WARNING: When you add a folder (as opposed to a file) to the list of keyfiles, only the path is remembered, not the filenames! This means e.g. that if you create a new file in the folder or if you copy an additional file to the folder, then all volumes that used keyfiles from the folder will be impossible to mount (until you remove the newly added file from the folder).

Empty Password & Keyfile

When a keyfile is used, the password may be empty, so the keyfile may become the only item necessary to mount the volume (which we do not recommend). If default keyfiles are set and enabled when mounting a volume, then before prompting for a password, TrueCrypt first automatically attempts to mount using an empty password plus default keyfiles. If you need to set Mount Options (e.g., mount as read-only, protect hidden volume etc.) for a volume being mounted this way, hold down the *Control* (*Ctrl*) key while clicking *Mount* (or select *Mount with Options* from the *Volumes* menu). This will open the *Mount Options* dialog.

Quick Selection

Keyfiles and keyfile search paths can be quickly selected in the following ways:

- Right-click the *Keyfiles* button in the password entry dialog window and select one of the menu items.
- Drag the corresponding file/folder icons to the keyfile dialog window or to the password entry dialog.

* Found at the time when you are mounting the volume, changing its password, or performing any other operation that involves re-encryption of the volume header.

Keyfiles -> Add/Remove Keyfiles to/from Volume

This function allows you to re-encrypt a volume header with a header encryption key derived from any number of keyfiles (with or without a password), or no keyfiles at all. Thus, a volume which is possible to mount using only a password can be converted to a volume that require keyfiles (in addition to the password) in order to be possible to mount. Note that the volume header contains the master encryption key with which the volume is encrypted. Therefore, the data stored on the volume will *not* be lost after you use this function.

This function can also be used to change/set volume keyfiles (i.e., to remove some or all keyfiles, and to apply new ones).

Remark: This function is internally equal to the Password Change function.

When TrueCrypt re-encrypts a volume header, the original volume header is first overwritten 200 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunneling microscopy [17] to recover the overwritten header (however, see also the chapter *Security Precautions*).

Keyfiles -> Remove All Keyfiles from Volume

This function allows you to re-encrypt a volume header with a header encryption key derived from a password and no keyfiles (so that it can be mounted using only a password, without any keyfiles). Note that the volume header contains the master encryption key with which the volume is encrypted. Therefore, the data stored on the volume will *not* be lost after you use this function.

Remark: This function is internally equal to the Password Change function.

When TrueCrypt re-encrypts a volume header, the original volume header is first overwritten 200 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunneling microscopy [17] to recover the overwritten header (however, see also the chapter *Security Precautions*).

Keyfiles -> Generate Random Keyfile

You can use this function to generate a file with random content, which you can use as a keyfile (recommended). This function uses the TrueCrypt Random Number Generator. Note that the resulting file size is always 64 bytes (i.e., 512 bits), which is also the maximum possible TrueCrypt password length.

Keyfiles -> Set Default Keyfile/Paths

Use this function to set default keyfiles and/or default keyfile search paths. This function is particularly useful if you, for example, store keyfiles on a USB memory stick that you carry with you. You can add its drive letter to the default keyfile configuration. To do so, click *Add Path*, browse to the drive letter assigned to the USB memory stick, and click *OK*. Now each time you mount a volume (and if *Use keyfiles* is checked in the password dialog), TrueCrypt will scan the path and use all files that it finds there as keyfiles.

WARNING: When you add a folder (as opposed to a file) to your default keyfile list, only the path is remembered, not the filenames! This means e.g. that if you create a new file in the folder or if you copy an additional file to the folder, then all volumes that used keyfiles from the folder will be impossible to mount (until you remove the newly added file from the folder).

IMPORTANT: Note that when you set default keyfiles and/or default keyfile search paths, the filenames and paths are saved unencrypted in the file Default Keyfiles.xml*. For more information, please see the chapter TrueCrypt System Files & Application Data.*

Security Tokens & Smart Cards

TrueCrypt supports security (or cryptographic) tokens and smart cards (smart card readers) that can be accessed using the PKCS #11 protocol [23]. For more information, please see the section *Security Tokens and Smart Cards* in the chapter *Keyfiles*.

Traveler Mode

TrueCrypt can run in so-called ‘traveler’ mode, which means that it does not have to be installed on the operating system under which it is run. However, there are two things to keep in mind:

1) You need administrator privileges in order to able to run TrueCrypt in ‘traveler’ mode.
2) After examining the registry file, it may be possible to tell that TrueCrypt was run (and that a TrueCrypt volume was mounted) on a Windows system even if it is run in traveler mode.

If you need to solve these problems, we recommend using *BartPE* for this purpose. For further information on *BartPE*, see the question “*Is it possible to use TrueCrypt without leaving any 'traces' on Windows?*” in the section *Frequently Asked Questions.*

There are two ways to run TrueCrypt in ‘traveler’ mode:

1) After you unpack the binary distribution archive, you can directly run *TrueCrypt.exe*.
2) You can use the *Traveler Disk Setup* facility to prepare a special ‘traveler’ disk and launch TrueCrypt from there.

The second option has several advantages, which are described in the following sections in this chapter.

Note: When running in 'traveler' mode, the TrueCrypt driver is unloaded when it is no longer needed (e.g., when all instances of the main application and/or of the Volume Creation Wizard are closed and no TrueCrypt volumes are mounted). However, if you force dismount on a TrueCrypt volume when TrueCrypt runs in ‘traveler’ mode, the TrueCrypt driver will *not* be unloaded when you exit TrueCrypt (it will be unloaded only when you shut down or restart the system). This prevents various problems caused by a bug in Windows (for instance, it would be impossible to start TrueCrypt again as long as there are applications using the dismounted volume).

Tools -> Traveler Disk Setup

You can use this facility to prepare a special ‘traveler’ disk and launch TrueCrypt from there. Note that TrueCrypt ‘traveler disk’ is *not* a TrueCrypt volume but an *unencrypted* volume. A ‘traveler disk’ contains TrueCrypt executable files and optionally the ‘autorun.inf’ script (see the section *AutoRun Configuration* below). After you select *Tools -> Traveler Disk Setup*, the *Traveler Disk Setup* dialog box should appear. Some of the parameters that can be set within the dialog deserve further explanation:

Include TrueCrypt Volume Creation Wizard

Check this option, if you need to create new TrueCrypt volumes using TrueCrypt run from the ‘traveler’ disk you will create. Unchecking this option saves space on the ‘traveler’ disk.

AutoRun Configuration (autorun.inf)

In this section, you can configure the ‘traveler disk’ to automatically start TrueCrypt or mount a specified TrueCrypt volume when the ‘traveler disk’ is inserted. This is accomplished by creating a special script file called ‘*autorun.inf*’ on the traveler disk. This file is automatically executed by the operating system each time the ‘traveler disk’ is inserted. Note that this feature only works for removable storage devices such as CD/DVD (Windows XP SP2 or Windows Vista is required for this feature to work on USB memory sticks) and only when it is enabled in the operating system. Also note that the ‘*autorun.inf*’ file must be in the root directory (i.e., for example *G:*\, *X:*\, or *Y:*\ etc.) of an **unencrypted** disk in order for this feature to work.

Using TrueCrypt Without Administrator Privileges

In Windows, a user who does not have administrator privileges *can* use TrueCrypt, but only after a system administrator installs TrueCrypt on the system. The reason for that is that TrueCrypt needs a device driver to provide transparent on-the-fly encryption/decryption, and users without administrator privileges cannot install/start device drivers in Windows.

After a system administrator installs TrueCrypt on the system, users without administrator privileges will be able to run TrueCrypt, mount/dismount any type of TrueCrypt volume, load/save data from/to it, and create file-hosted TrueCrypt volumes on the system. However, users without administrator privileges cannot encrypt/format partitions, cannot create NTFS volumes, cannot install/uninstall TrueCrypt, cannot change passwords/keyfiles for TrueCrypt partitions/devices, cannot backup/restore headers of TrueCrypt partitions/devices, and they cannot run TrueCrypt in 'traveler' mode.

TrueCrypt Background Task

When the main TrueCrypt window is closed, the TrueCrypt Background Task takes care of the following tasks/functions:

1) Hot keys
2) Auto-dismount (e.g., upon log off, inadvertent host device removal, time-out, etc.)
3) Notifications (e.g., when damage to hidden volume is prevented)
4) Tray icon

WARNING: If neither the TrueCrypt Background Task nor TrueCrypt is running, the above-mentioned tasks/functions are disabled.

The TrueCrypt Background Task is actually the *TrueCrypt.exe* application, which continues running in the background after you close the main TrueCrypt window. Whether it is running or not can be determined by looking at the system tray area. If you can see the TrueCrypt icon there, then the TrueCrypt Background Task is running. You can click the icon to open the main TrueCrypt window. Right-click on the icon opens a popup menu with various TrueCrypt-related functions.

You can shut down the Background Task at any time by right-clicking the TrueCrypt tray icon and selecting *Exit*. If you need to disable the TrueCrypt Background Task completely and permanently, select *Settings -> Preferences* and uncheck the option *Enabled* in the *TrueCrypt Background Task* area of the *Preferences* dialog window.

Language Packs

Language packs contain third-party translations of the TrueCrypt user interface texts. Some language packs also contain translated TrueCrypt User Guide. Note that language packs are currently supported only by the Windows version of TrueCrypt.

Installation

To install a language pack, follow these steps:

1. Download a language pack from: http://www.truecrypt.org/localizations.php
2. Exit TrueCrypt (if it is running).
3. Extract the language pack to the folder to which you installed TrueCrypt, i.e. the folder in which the file 'TrueCrypt.exe' resides; for example, 'C:\Program Files\TrueCrypt' or 'C:\Program Files (x86)\TrueCrypt', etc.
4. Run TrueCrypt.
5. The language pack should be automatically detected, loaded, and set as the default language pack. (You can select a language at any time by clicking *Settings -> Language*).

To revert to English, select *Settings -> Language*. Then select *English* and click *OK*.

Encryption Algorithms

TrueCrypt volumes can be encrypted using the following algorithms:

Algorithm	Designer(s)	Key Size (Bits)	Block Size (Bits)	Mode of Operation
AES	J. Daemen, V. Rijmen	256	128	XTS
Serpent	R. Anderson, E. Biham, L. Knudsen	256	128	XTS
Twofish	B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson	256	128	XTS
AES-Twofish		256; 256	128	XTS
AES-Twofish-Serpent		256; 256; 256	128	XTS
Serpent-AES		256; 256	128	XTS
Serpent-Twofish-AES		256; 256; 256	128	XTS
Twofish-Serpent		256; 256	128	XTS

For information about XTS mode, please see the section *Modes of Operation*.

AES

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm (Rijndael, designed by Joan Daemen and Vincent Rijmen, published in 1998) that may be used by US federal departments and agencies to cryptographically protect sensitive information [3]. TrueCrypt uses AES with 14 rounds and a 256-bit key (i.e., AES-256, published in 2001) operating in XTS mode (see the section *Modes of Operation*).

In June 2003, after the NSA (US National Security Agency) conducted a review and analysis of AES, the U.S. CNSS (Committee on National Security Systems) announced in [1] that the design and strength of AES-256 (and AES-192) are sufficient to protect classified information up to the Top Secret level. This is applicable to all U.S. Government Departments or Agencies that are considering the acquisition or use of products incorporating the Advanced Encryption Standard (AES) to satisfy Information Assurance requirements associated with the protection of national security systems and/or national security information [1].

Serpent

Designed by Ross Anderson, Eli Biham, and Lars Knudsen; published in 1998. It uses a 256-bit key, 128-bit block, and operates in XTS mode (see the section *Modes of Operation*). Serpent was one of the AES finalists. It was not selected as the proposed AES algorithm even though it appeared to have a higher security margin than the winning Rijndael [4]. More concretely, Serpent appeared to have a *high* security margin, while Rijndael appeared to have only an *adequate* security margin [4]. Rijndael has also received some criticism suggesting that its mathematical structure might lead to attacks in the future [4].

In [5], the Twofish team presents a table of safety factors for the AES finalists. Safety factor is defined as: number of rounds of the full cipher divided by the largest number of rounds that has been broken. Hence, a broken cipher has the lowest safety factor 1. Serpent had the highest safety factor of the AES finalists: 3.56 (for all supported key sizes). Rijndael-256 had a safety factor of 1.56.

In spite of these facts, Rijndael was considered an appropriate selection for the AES for its combination of security, performance, efficiency, implementability, and flexibility [4]. At the last AES Candidate Conference, Rijndael got 86 votes, Serpent got 59 votes, Twofish got 31 votes, RC6 got 23 votes, and MARS got 13 votes [18, 19].*

Twofish

Designed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson; published in 1998. It uses a 256-bit key and 128-bit block and operates in XTS mode (see the section *Modes of Operation*). Twofish was one of the AES finalists. This cipher uses key-dependent S-boxes. Twofish may be viewed as a collection of 2^{128} different cryptosystems, where 128 bits derived from a 256-bit key control the selection of the cryptosystem [4]. In [13], the Twofish team asserts that key-dependent S-boxes constitute a form of security margin against unknown attacks [4].

AES-Twofish

Two ciphers in a cascade [15, 16] operating in XTS mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with Twofish (256-bit key) in XTS mode and then with AES (256-bit key) in XTS mode. Each of the cascaded ciphers uses its own key. All encryption keys are mutually independent (note that header keys are independent as well, even though they are derived from a single password – see *Header Key Derivation, Salt, and Iteration Count*). See above for information on the individual cascaded ciphers.

* These are positive votes. If negative votes are subtracted from the positive votes, the following results are obtained: Rijndael: 76 votes, Serpent: 52 votes, Twofish: 10 votes, RC6: -14 votes, MARS: -70 votes [19].

AES-Twofish-Serpent

Three ciphers in a cascade [15, 16] operating in XTS mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with Serpent (256-bit key) in XTS mode, then with Twofish (256-bit key) in XTS mode, and finally with AES (256-bit key) in XTS mode. Each of the cascaded ciphers uses its own key. All encryption keys are mutually independent (note that header keys are independent as well, even though they are derived from a single password – see the section *Header Key Derivation, Salt, and Iteration Count*). See above for information on the individual cascaded ciphers.

Serpent-AES

Two ciphers in a cascade [15, 16] operating in XTS mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with AES (256-bit key) in XTS mode and then with Serpent (256-bit key) in XTS mode. Each of the cascaded ciphers uses its own key. All encryption keys are mutually independent (note that header keys are independent as well, even though they are derived from a single password – see the section *Header Key Derivation, Salt, and Iteration Count*). See above for information on the individual cascaded ciphers.

Serpent-Twofish-AES

Three ciphers in a cascade [15, 16] operating in XTS mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with AES (256-bit key) in XTS mode, then with Twofish (256-bit key) in XTS mode, and finally with Serpent (256-bit key) in XTS mode. Each of the cascaded ciphers uses its own key. All encryption keys are mutually independent (note that header keys are independent as well, even though they are derived from a single password – see the section *Header Key Derivation, Salt, and Iteration Count*). See above for information on the individual cascaded ciphers.

Twofish-Serpent

Two ciphers in a cascade [15, 16] operating in XTS mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with Serpent (256-bit key) in XTS mode and then with Twofish (256-bit key) in XTS mode. Each of the cascaded ciphers uses its own key. All encryption keys are mutually independent (note that header keys are independent as well, even though they are derived from a single password – see the section *Header Key Derivation, Salt, and Iteration Count*). See above for information on the individual cascaded ciphers.

Hash Algorithms

In the Volume Creation Wizard, in the password change dialog window, and in the Keyfile Generator dialog window, you can select a hash algorithm. A user-selected hash algorithm is used by the TrueCrypt Random Number Generator as a pseudorandom “mixing” function, and by the header key derivation function (HMAC based on a hash function, as specified in PKCS #5 v2.0) as a pseudorandom function. When creating a new volume, the Random Number Generator generates the master key, secondary key (XTS mode), and salt. For more information, please see the section *Random Number Generator* and section *Header Key Derivation, Salt, and Iteration Count*.

RIPEMD-160

RIPEMD-160, published in 1996, is a hash algorithm designed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel in an open academic community. The size of the output of RIPEMD-160 is 160 bits. RIPEMD-160 is a strengthened version of the RIPEMD hash algorithm that was developed in the framework of the European Union’s project RIPE (*RACE Integrity Primitives Evaluation*), 1988-1992. RIPEMD-160 was adopted by the International Organization for Standardization (ISO) and the IEC in the ISO/IEC 10118-3:2004 international standard [21].

SHA-512

SHA-512 is a hash algorithm designed by the NSA and published by NIST in FIPS PUB 180-2 [14] in 2002 (the first draft was published in 2001). The size of the output of this algorithm is 512 bits.

Whirlpool

The Whirlpool hash algorithm was designed by Vincent Rijmen (co-designer of the AES encryption algorithm) and Paulo S. L. M. Barreto. The size of the output of this algorithm is 512 bits. The first version of Whirlpool, now called Whirlpool-0, was published in November 2000. The second version, now called Whirlpool-T, was selected for the NESSIE (*New European Schemes for Signatures, Integrity and Encryption*) portfolio of cryptographic primitives (a project organized by the European Union, similar to the AES competition). TrueCrypt uses the third (final) version of Whirlpool, which was adopted by the International Organization for Standardization (ISO) and the IEC in the ISO/IEC 10118-3:2004 international standard [21].

Supported Operating Systems

This version of TrueCrypt supports the following operating systems:

- Windows Vista
- Windows Vista x64 (64-bit) Edition
- Windows XP
- Windows XP x64 (64-bit) Edition
- Windows Server 2008
- Windows Server 2008 x64 (64-bit)
- Windows Server 2003
- Windows Server 2003 x64 (64-bit)
- Windows 2000 SP4

- Mac OS X 10.5 Leopard
- Mac OS X 10.4 Tiger

- Linux (kernel 2.4, 2.6 or compatible)

Note: The following operating systems (among others) are not supported: Windows 2003 IA-64, Windows 2008 IA-64, Windows XP IA-64, Windows 95/98/ME/NT.

See also the section *Operating Systems Supported for System Encryption*.

Command Line Usage

Note that this section applies to the Windows version of TrueCrypt. For information on command line usage applying to the **Linux and Mac OS X versions**, please run: `truecrypt -h`

/help or */?*	Display command line help.
/volume or */v*	File and path name of a TrueCrypt volume to mount (do not use when dismounting). To mount a hard disk partition, use, for example, `/v \Device\Harddisk1\Partition3` (to determine the path to a partition, run TrueCrypt and click *Select Device*). Note that device paths are case-sensitive.
/letter or */l*	Driver letter to mount the volume as. When `/l` is omitted and when `/a` is used, the first free drive letter is used.
/explore or */e*	Open an Explorer window after a volume has been mounted.
/beep or */b*	Beep after a volume has been successfully mounted or dismounted.
/auto or */a*	If no parameter is specified, automatically mount the volume. If `devices` is specified as the parameter (e.g., `/a devices`), auto-mount all currently accessible device/partition-hosted TrueCrypt volumes. If `favorites` is specified as the parameter, auto-mount favorite volumes. Note that `/auto` is implicit if `/quit` and `/volume` are specified.
/dismount or */d*	Dismount volume specified by drive letter (e.g., `/d x`). When no drive letter is specified, dismounts all currently mounted TrueCrypt volumes.
/force or */f*	Forces dismount (if the volume to be dismounted contains files being used by the system or an application) and forces mounting in shared mode (i.e., without exclusive access).
/keyfile or */k*	Specifies a keyfile or a keyfile search path. For multiple keyfiles, specify e.g.: `/k c:\keyfile1.dat /k d:\KeyfileFolder /k c:\kf2` To specify a keyfile stored on a security token or smart card, use the following syntax: `token://slot/`*SLOT_NUMBER*`/file/`*FILE_NAME*
/tokenlib	Use the specified PKCS #11 library for security tokens and smart cards.
/cache or */c*	`y` or no parameter: enable password cache; `n`: disable password cache (e.g., `/c n`). Note that turning the password cache off will not clear it (use `/w` to clear the password cache).
/history or */h*	`y` or no parameter: enables saving history of mounted volumes; `n`: disables saving history of mounted volumes (e.g., `/h n`).
/wipecache or */w*	Wipes any passwords cached in the driver memory.
/password or */p*	The volume password. If the password contains spaces, it must be enclosed in quotation marks (e.g., `/p "My Password"`). Use `/p ""` to

specify an empty password. *Warning: This method of entering a volume password may be insecure, for example, when an unencrypted command prompt history log is being saved to unencrypted disk.*

/quit or */q* — Automatically perform requested actions and exit (main TrueCrypt window will not be displayed). If `preferences` is specified as the parameter (e.g., `/q preferences`), then program settings are loaded/saved and they override settings specified on the command line.
`/q background` launches the TrueCrypt Background Task (tray icon).

/silent or */s* — If `/q` is specified, suppresses interaction with the user (prompts, error messages, warnings, etc.). If `/q` is not specified, this option has no effect.

/mountoption or */m* — `ro` or `readonly`: Mount volume as read-only.

`rm` or `removable`: Mount volume as removable medium.

`ts` or `timestamp`: Do not preserve container timestamps

`sm` or `system`: Without pre-boot authentication, mount a partition that is within the key scope of system encryption (for example, a partition located on the encrypted system drive of another operating system that is not running). Useful e.g. for backup or repair operations.
Note: If you supply a password as a parameter of `/p`, make sure that the password has been typed using the standard US keyboard layout (in contrast, the GUI ensures this automatically).

`bk` or `headerbak`: Mount volume using embedded backup header.
Note: All volumes created by TrueCrypt 6.0 or later contain an embedded backup header (located at the end of the volume).

`recovery`: Do not verify any checksums stored in the volume header. This option should be used only when the volume header is damaged and the volume cannot be mounted even with the mount option `headerbak`.

Example: `/m ro`. To specify multiple mount options, use e.g.: `/m rm /m ts`

TrueCrypt Format.exe (TrueCrypt Volume Creation Wizard):

/noisocheck or */n* — Do not verify that TrueCrypt Rescue Disks are correctly burned. This can be useful e.g. in corporate environments where it may be more convenient to maintain a central repository of ISO images rather than a repository of CDs or DVDs. **WARNING**: *Never attempt to use this option to facilitate the reuse of a previously created TrueCrypt Rescue Disk.* Note that every time you encrypt a system partition/drive, you must create a new TrueCrypt Rescue Disk even if you use the same password. A previously created TrueCrypt Rescue Disk cannot be reused because it was created for a different master key.

Syntax

`TrueCrypt.exe [/a [devices|favorites]] [/b] [/c [y|n]] [/d [`*drive letter*`]] [/e] [/f] [/h [y|n]] [/k` *keyfile or search path*`] [/l` *drive letter*`] [/m {bk|rm|recovery|ro|sm|ts}] [/p` *password*`] [/q [background|preferences]] [/s] [/tokenlib` *path*`] [/v` *volume*`] [/w]`

`"TrueCrypt Format.exe" [/n]`

Note that the order in which options are specified does not matter.

Examples

Mount the volume *d:\myvolume* as the first free drive letter, using the password prompt (the main program window will not be displayed):

```
truecrypt /q /v d:\myvolume
```

Dismount a volume mounted as the drive letter *X* (the main program window will not be displayed):

```
truecrypt /q /dx
```

Mount a volume called *myvolume.tc* using the password *MyPassword*, as the drive letter *X*. TrueCrypt will open an explorer window and beep, mounting will be automatic:

```
truecrypt /v myvolume.tc /lx /a /p MyPassword /e /b
```

Sharing over Network

If there is a need to access a single TrueCrypt volume simultaneously from multiple operating systems, there are two options:

1. A TrueCrypt volume is mounted only on a single computer (for example, on a server) and only the content of the mounted TrueCrypt volume (i.e., the file system within the TrueCrypt volume) is shared over a network. Users on other computers or systems will not mount the volume (it is already mounted on the server).

   Advantages: All users can write data to the TrueCrypt volume. The shared volume may be both file-hosted and partition/device-hosted.

   Disadvantage: Data sent over the network will not be encrypted. However, it is still possible to encrypt them using e.g. SSL, TLS, VPN, or other technologies.

2. A dismounted TrueCrypt file container is stored on a single computer (for example, on a server). This encrypted file is shared over a network. Users on other computers or systems will locally mount the shared file. Thus, the volume will be mounted simultaneously under multiple operating systems.

   Advantage: Data sent over the network will be encrypted (however, it is still recommended to encrypt them using e.g. SSL, TLS, VPN, or other appropriate technologies to make traffic analysis more difficult and to preserve the integrity of the data).

   Disadvantages: The shared volume may be only file-hosted (not partition/device-hosted). The volume must be mounted in read-only mode under each of the systems (see the section *Mount Options* for information on how to mount a volume in read-only mode). Note that this requirement applies to unencrypted volumes as well. One of the reasons is, for example, the fact that data read from a conventional file system under one OS while the file system is being modified by another OS might be inconsistent (which could result in data corruption).

Security Precautions

This chapter informs about things that might affect the security of sensitive data stored on TrueCrypt volumes. Please note that it is impossible to inform about *all* security risks here. There are, unfortunately, too many of them and it would require thousands of pages to describe them.

Paging File

*Note: The issue described below does **not** affect you if the system partition or system drive is encrypted (for more information, see the chapter* System Encryption*) and if all paging files are located on one or more of the partitions within the key scope of system encryption (which they typically are, by default), for example, on the partition where Windows is installed.*

Also called ‘swap file’; Windows uses this file (usually stored on a hard drive) to hold parts of programs and data files that do not fit in memory. This means that sensitive data, which you believe are only stored in RAM, can actually be written *unencrypted* to a hard drive by Windows without you knowing.

TrueCrypt always attempts to lock the memory areas in which cached passwords, encryption keys, and other sensitive data are stored, in order to prevent such data from being leaked to paging files. However, note that Windows may reject or fail to lock memory for various (documented and undocumented) reasons. Furthermore, TrueCrypt *cannot* prevent the contents of sensitive files that are opened in RAM from being saved *unencrypted* to a paging file (note that when you open a file stored on a TrueCrypt volume, for example, in a text editor, then the content of the file is stored *unencrypted* in RAM).

Solution 1: Encrypt the system partition/drive (for information on how to do so, see the chapter *System Encryption*) and make sure that all paging files are located on one or more of the partitions within the key scope of system encryption (which they typically are, by default), for example, on the partition where Windows is installed.

Note: You may also want to consider creating a hidden operating system (for more information, see the section Hidden Operating System*).*

Solution 2: Disable the paging file feature, at least for each session during which you work with sensitive data and during which you mount TrueCrypt volumes. To do so, right-click the '*Computer*' (or '*My Computer*') icon on the desktop or in the *Start Menu*, and then select *Properties* -> (Windows Vista only: -> *Advanced System Settings* ->) *Advanced* tab -> section *Performance* -> *Settings* -> *Advanced* tab -> section *Virtual memory* -> *Change* -> *No paging file* -> *Set* -> *OK* ; finally, restart the computer. Note: To our best knowledge, Windows 2000 users cannot disable the paging file feature completely. Therefore, we recommend that TrueCrypt users upgrade from Windows 2000 to a newer version of Windows.

Hibernation File

Note: The issue described below does not affect you if the system partition or system drive is encrypted[*] (for more information, see the chapter System Encryption*) and if the hibernation file is located on one the partitions within the key scope of system encryption (which it typically is, by default), for example, on the partition where Windows is installed. When the computer hibernates, data are encrypted on the fly before they are written to the hibernation file.*

When a computer hibernates (or enters a power-saving mode), the content of its system memory is written to a so-called hibernation file on the hard drive. By default, before a computer hibernates (or enters a power-saving mode), TrueCrypt automatically dismounts all mounted TrueCrypt volumes, erases their master keys stored in RAM, and cached passwords (stored in RAM) if there are any. Keep in mind, however, that if you do not use system encryption (see the chapter *System Encryption*), TrueCrypt cannot reliably prevent the contents of sensitive files opened in RAM from being saved unencrypted to a hibernation file. Note that when you open a file stored on a TrueCrypt volume, for example, in a text editor, then the content of the file is stored unencrypted in RAM (and it may remain unencrypted in RAM until the computer is turned off).

Solution 1*:* Encrypt the system partition/drive (for information on how to do so, see the chapter *System Encryption*) and make sure that the hibernation file is located on one of the partitions within the key scope of system encryption (which it typically is, by default), for example, on the partition where Windows is installed. When the computer hibernates, data will be encrypted on the fly before they are written to the hibernation file.

Note: You may also want to consider creating a hidden operating system (for more information, see the section Hidden Operating System*).*

Solution 2: If you do not use system encryption, disable or prevent hibernation on your computer at least for each session during which you work with any sensitive data and during which you mount a TrueCrypt volume.

Memory Dump Files

Note: The issue described below does ***not*** *affect you if the system partition or system drive is encrypted (for more information, see the chapter* System Encryption*) and if the system is configured to write memory dump files to the system drive (which it typically is, by default).*

Most operating systems, including Windows, can be configured to write debugging information and contents of the system memory to so-called memory dump files when an error occurs (system crash, "blue screen," bug check). Therefore, memory dump files may contain sensitive data. TrueCrypt *cannot* prevent cached passwords, encryption keys, and the contents of sensitive files opened in RAM from being saved *unencrypted* to memory dump files. Note that when you open a file stored on a TrueCrypt volume, for example, in a text editor, then the content of the file is stored *unencrypted* in RAM (and it may remain *unencrypted* in RAM until the computer is turned off). Also note that when a TrueCrypt volume is mounted, its master key is stored *unencrypted* in RAM. Therefore, we strongly recommend that you disable memory dump file generation on your

[*] Disclaimer: As Microsoft does not provide any API for handling hibernation, non-Microsoft developers of disk encryption software are forced to modify undocumented components of Windows in order to allow users to encrypt hibernation files. Therefore, no disk encryption software (except for Microsoft's BitLocker) can currently guarantee that hibernation files will always be encrypted. At anytime, Microsoft can arbitrarily modify components of Windows (using the Auto Update feature of Windows) that are not publicly documented or accessible via a public API. Any such change, or the use of an untypical or custom storage device driver, may cause any non-Microsoft disk encryption software to fail to encrypt the hibernation file.

computer at least for each session during which you work with any sensitive data and during which you mount a TrueCrypt volume. To do so in Windows XP/Vista, right-click the '*Computer*' (or '*My Computer*') icon on the desktop or in the *Start Menu*, and then select *Properties* -> (Windows Vista only: -> *Advanced System Settings* ->) *Advanced* tab -> section *Startup and Recovery* -> *Settings* -> section *Write debugging information* -> select *(none)* -> *OK.*

Note: If the system partition/drive is encrypted by TrueCrypt and if the system is configured to write memory dump files to the system drive (which it typically is, by default), the TrueCrypt driver automatically prevents Windows from writing any data to memory dump files (for information on how to encrypt the system partition/drive, see the chapter System Encryption).

Multi-User Environment

Keep in mind, that the content of a mounted TrueCrypt volume is visible (accessible) to all logged on users (NTFS file permissions can be configured to prevent this). Also note that switching users in Windows XP/Vista (*Fast User Switching* functionality) does *not* dismount a successfully mounted TrueCrypt volume (unlike system restart, which dismounts all mounted TrueCrypt volumes).

On Windows 2000, the container file permissions are ignored when a file-hosted TrueCrypt volume is to be mounted. On all supported versions of Windows, users without administrator privileges can mount any partition/device-hosted TrueCrypt volume (provided that he or she supplies the correct password and/or keyfiles). However, a user without administrator privileges can dismount only volumes that he or she mounted.

Unencrypted Data in RAM

It is important to note that TrueCrypt is *disk* encryption software, which encrypts only disks, not RAM (memory).

Keep in mind that most programs do not clear the memory area (buffers) in which they store unencrypted (portions of) files they load from a TrueCrypt volume. This means that after you exit such a program, unencrypted data it worked with may remain in memory (RAM) until the computer is turned off (and, according to some researchers, even for some time after the power is turned off[*]). Also note that if you open a file stored on a TrueCrypt volume, for example, in a text editor and then force dismount on the TrueCrypt volume, then the file will remain unencrypted in the area of memory (RAM) used by (allocated to) the text editor. This applies to forced auto-dismount as well.

Inherently, unencrypted master keys have to be stored in RAM as well. When a non-system TrueCrypt volume is dismounted, TrueCrypt erases its master keys (stored in RAM). When the computer is cleanly restarted (or cleanly shut down) or hibernates, all non-system TrueCrypt volumes are automatically dismounted (by default) and, thus, all master keys stored in RAM are erased by the TrueCrypt driver (except master keys for system partitions/drives — see below). However, when power supply is abruptly interrupted, when the computer is reset (not cleanly restarted), or when the system crashes, **TrueCrypt naturally stops running and therefore cannot** erase any keys or any other sensitive data. Furthermore, as Microsoft does not provide any appropriate API for handling hibernation and shutdown, master keys used for system encryption cannot be reliably (and are not) erased from RAM when a computer hibernates, is shut

[*] Allegedly, for 1.5-35 seconds under normal operating temperatures (26-44 °C) and up to several hours when the memory modules are cooled (when the computer is running) to very low temperatures (e.g. -50 °C). New types of memory modules allegedly exhibit a much shorter decay time (e.g. 1.5-2.5 seconds) than older types.

down or restarted.*

To summarize, TrueCrypt **cannot** and does **not** ensure that RAM contains no sensitive data (e.g. passwords, master keys, or decrypted data). Therefore, after each session in which you work with a TrueCrypt volume or in which an encrypted operating system is running, you must shut down or hibernate the computer and then leave it powered off for several minutes before turning it on again. This is required to clear the RAM (see also the section *Hibernation File*).

Changing Passwords and Keyfiles

Note that the volume header (which is encrypted with a header key derived from a password/keyfile) contains the master key with which the volume is encrypted. If an adversary is allowed to make a copy of your volume before you change the volume password and/or keyfile(s), he may be able to use his copy or fragment (the old header) of the TrueCrypt volume to mount your volume using a compromised password (for example, captured by a keystroke logger) and/or compromised keyfiles that were necessary to mount the volume before you changed the volume password and/or keyfile(s).

If you are not sure whether an adversary knows your password (or has your keyfiles) and whether he has a copy of your volume when you need to change its password and/or keyfiles, it is strongly recommended that you create a new TrueCrypt volume and move files from the old volume to the new volume (the new volume will have a different master key).

Also note that if an adversary knows your password (or has your keyfiles) and has access to your volume, he may be able to retrieve and keep its master key. If he does, he may be able to decrypt your volume even after you change its password and/or keyfile(s) (because the master key does not change when you change the volume password and/or keyfiles). In such a case, create a new TrueCrypt volume and move all files from the old volume to this new one.

Data Leaks

*Note: The issues described below do **not** affect you if the system partition or system drive is encrypted (for more information, see the chapter* System Encryption*) and if no unencrypted filesystems (or volumes) are accessible during a session in which a TrueCrypt volume is mounted or in which an encrypted operating system is running.*

When a TrueCrypt volume is mounted, the operating system and third-party applications may write to unencrypted volumes (typically, to the unencrypted system volume) unencrypted information about the data stored in the TrueCrypt volume (e.g. filenames and locations of recently accessed files, databases created by file indexing tools, etc.), or the data itself in an unencrypted form (temporary files, etc.), or unencrypted information about the filesystem residing in the TrueCrypt volume. Note that Windows automatically records large amounts of potentially sensitive data, such as the names and locations of files you open, applications you run, etc.

* Before a key can be erased from RAM, the corresponding TrueCrypt volume must be dismounted. For non-system volumes, this does not cause any problems. However, as Microsoft currently does not provide any appropriate API for handling the final phase of the system shutdown process, paging files located on encrypted system volumes that are dismounted during the system shutdown process may still contain valid swapped-out memory pages (including portions of Windows system files). This can cause 'blue screen' errors. Therefore, TrueCrypt does not dismount encrypted system volumes and consequently cannot clear the master keys of the system volumes when the system is shut down or restarted.

Solution: Encrypt the system partition/drive (for information on how to do so, see the chapter *System Encryption*) and make sure that only encrypted filesystems are mounted during a session in which you work with sensitive data (unencrypted filesystems/volumes should not be mounted/accessible during a session in which a TrueCrypt volume is mounted or an encrypted operating system is running).

Note: You may also want to consider creating a hidden operating system (for more information, see the section Hidden Operating System*).*

Windows Registry

It is important to note that TrueCrypt provides plausible deniability *only* in the sense that it is impossible to prove that a file or a partition is a TrueCrypt volume and that a hidden TrueCrypt volume exists (see the chapter *Plausible Deniability*). Windows stores various data in the registry file, which TrueCrypt cannot securely and reliably erase. After examining the registry file, the attacker may be able to tell that TrueCrypt was run on the system, that a TrueCrypt volume was mounted (but he cannot tell/determine what the location/filename/size/type* of the volume was) and which drive letters have been used for TrueCrypt volume(s) (but he cannot determine the locations/filenames/sizes/types of the volumes).

Note*: You can encrypt the registry file by encrypting the system partition/drive (for information on how to do so, see the chapter* System Encryption*). You may also want to consider creating a hidden operating system (for more information, see the section* Hidden Operating System*).*

Wear-Leveling

Some storage devices (e.g., some USB flash drives) and some file systems utilize so-called wear-leveling mechanisms to extend the lifetime of the storage device or medium. These mechanisms ensure that even if an application repeatedly writes data to the same logical sector, the data is distributed evenly across the medium (logical sectors are remapped to different physical sectors). Therefore, multiple "versions" of a single sector may be available to an attacker. This may have various security implications. For instance, when you change a volume password/keyfile(s), the volume header is, under normal conditions, overwritten with a re-encrypted version of the header. However, when the volume resides on a device that utilizes a wear-leveling mechanism, TrueCrypt cannot ensure that the older header is really overwritten. If an adversary found the old volume header (which was to be overwritten) on the device, he could use it to mount the volume using an old compromised password (and/or using compromised keyfiles that were necessary to mount the volume before the volume header was re-encrypted). Due to security reasons, we recommend that TrueCrypt volumes are not created on devices (or in file systems) that utilize a wear-leveling mechanism. If you decide not to follow this recommendation and you intend to use system encryption (see the chapter *System Encryption*) when the system drive utilizes wear-leveling mechanisms, make sure the system partition/drive does not contain any sensitive data before you fully encrypt it (TrueCrypt cannot reliably perform secure in-place encryption of existing data on such a drive; however, after the system partition/drive has been fully encrypted, any new data that will be saved to it will be reliably encrypted on the fly). To find out whether a device utilizes a wear-leveling mechanism, please refer to documentation supplied with the device or contact the vendor/manufacturer.

* 'Type of volume' refers to whether it is a hidden or standard volume.

Reallocated Sectors

Some storage devices, such as hard drives, internally reallocate/remap bad sectors. Whenever the device detects a sector to which data cannot be written, it marks the sector as bad and remaps it to a sector in a hidden reserved area on the drive. Any subsequent read/write operations from/to the bad sector are redirected to the sector in the reserved area. This means that any existing data in the bad sector remains on the drive and it cannot be erased (overwritten with other data). This may have various security implications. For instance, data that is to be encrypted in place may remain unencrypted in the bad sector. Likewise, data to be erased (for example, during the process of creation of a hidden operating system) may remain in the bad sector. Additional examples of possible security implications are listed in the section *Wear-Leveling*. Please note that this list is not exhaustive (these are just examples). Also note that TrueCrypt *cannot* prevent any security issues related to or caused by reallocated sectors. To find out the number of reallocated sectors on a hard drive, you can use e.g. a third-party software tool for reading so-called S.M.A.R.T. data.

Defragmenting

When you defragment the file system in which you store a file-hosted TrueCrypt container, a copy of the TrueCrypt container (or of its fragment) may remain in the free space on the host volume (in the defragmented file system). This may have various security implications. For example, if you change the volume password/keyfile(s) afterwards, and an adversary finds the old copy or fragment (the old header) of the TrueCrypt volume, he might use it to mount the volume using an old compromised password (and/or using compromised keyfiles that were necessary to mount the volume before the volume header was re-encrypted). To prevent this, do one of the following:

- Use a partition/device-hosted TrueCrypt volume instead of file-hosted.
- Securely erase free space on the host volume (in the defragmented file system) after defragmenting.
- Do not defragment file systems in which you store TrueCrypt volumes.

Journaling File Systems

When a file-hosted TrueCrypt container is stored in a journaling file system (such as NTFS), a copy of the TrueCrypt container (or of its fragment) may remain in the free space on the host volume. This may have various security implications. For example, if you change the volume password/keyfile(s) and an adversary finds the old copy or fragment (the old header) of the TrueCrypt volume, he might use it to mount the volume using an old compromised password (and/or using compromised keyfiles using an old compromised password (and/or using compromised keyfiles that were necessary to mount the volume before the volume header was re-encrypted). Some journaling file systems also internally record file access times and other potentially sensitive information. To prevent possible security issues related to journaling file systems, do one the following:

- Use a partition/device-hosted TrueCrypt volume instead of file-hosted.
- Store the container in a non-journaling file system (for example, FAT32).

See also the subsection *Security Precautions Pertaining to Hidden Volumes* in the chapter *Plausible Deniability.*

How to Back Up Securely

Due to hardware or software errors/malfunctions, files stored on a TrueCrypt volume may become corrupted. Therefore, we strongly recommend that you backup all your important files regularly (this, of course, applies to any important data, not just to encrypted data stored on TrueCrypt volumes).

Non-System Volumes

To back up a non-system TrueCrypt volume securely, it is recommended to follow these steps:

1. Create a new TrueCrypt volume using the TrueCrypt Volume Creation Wizard (do not enable the *Quick Format* option or the *Dynamic* option). It will be your *backup* volume so its size should match (or be greater than) the size of your *main* volume.

   If the *main* volume is a hidden TrueCrypt volume (see the section *Hidden Volume*), the *backup* volume must be a hidden TrueCrypt volume as well. Before you create the hidden *backup* volume, you must create a new host (outer) volume for it without enabling the *Quick Format* option. In addition, especially if the *backup* volume is file-hosted, the hidden *backup* volume should occupy only a very small portion of the container and the outer volume should be almost completely filled with files (otherwise, the plausible deniability of the hidden volume might be adversely affected).

2. Mount the newly created *backup* volume.

3. Mount the *main* volume.

4. Copy all files from the mounted *main* volume directly to the mounted *backup* volume.

IMPORTANT: If you store the backup volume in any location that an adversary can repeatedly access (for example, on a device kept in a bank's safe deposit box), you should repeat all of the above steps (including the step 1) each time you want to back up the volume (see below).

If you follow the above steps, you will help prevent adversaries from finding out:

- Which sectors of the volumes are changing (because you always follow step 1). This is particularly important, for example, if you store the backup volume on a device kept in a bank's safe deposit box (or in any other location that an adversary can repeatedly access) and the volume contains a hidden volume (for more information, see the subsection *Security Precautions Pertaining to Hidden Volumes* in the chapter *Plausible Deniability*).

- That one of the volumes is a backup of the other.

System Partitions

Note: In addition to backing up files, we recommend that you also back up your TrueCrypt Rescue Disk (select *System* > *Create Rescue Disk*). For more information, see the section *TrueCrypt Rescue Disk*.

To back up an encrypted system partition securely and safely, it is recommended to follow these steps:

1. If you have multiple operating systems installed on your computer, boot the one that does not require pre-boot authentication.

   If you do not have multiple operating systems installed on your computer, you can boot a BartPE CD/DVD ('live' Windows entirely stored on and booted from a CD/DVD; for more information, search the section *Frequently Asked Questions* for the keyword 'BartPE').

   If none of the above is possible, connect your system drive as a secondary drive to another computer and then boot the operating system installed on the computer.

   Note: For security reasons, if the operating system that you want to back up resides in a hidden TrueCrypt volume (see the section *Hidden Operating System*), then the operating system that you boot in this step must be either another hidden operating system or a "live-CD" operating system (see above). For more information, see the subsection *Security Precautions Pertaining to Hidden Volumes* in the chapter *Plausible Deniability*.

2. Create a new non-system TrueCrypt volume using the TrueCrypt Volume Creation Wizard (do not enable the *Quick Format* option or the *Dynamic* option). It will be your *backup* volume so its size should match (or be greater than) the size of the system partition that you want to back up.

   If the operating system that you want to back up is installed in a hidden TrueCrypt volume (see the section *Hidden Operating System*), the *backup* volume must be a hidden TrueCrypt volume as well. Before you create the hidden *backup* volume, you must create a new host (outer) volume for it without enabling the *Quick Format* option. In addition, especially if the *backup* volume is file-hosted, the hidden *backup* volume should occupy only a very small portion of the container and the outer volume should be almost completely filled with files (otherwise, the plausible deniability of the hidden volume might be adversely affected).

3. Mount the newly created *backup* volume.

4. Mount the system partition that you want to back up by following these steps:

   a. Click *Select Device* and then select the system partition that you want to back up (in case of a hidden operating system, select the partition containing the hidden volume in which the operating system is installed).

   b. Click *OK*.

   c. Select *System* > *Mount Without Pre-Boot Authentication*.

   d. Enter your pre-boot authentication password and click *OK*.

5. Mount the *backup* volume and then copy all files from the system partition (mounted as a regular TrueCrypt volume since the previous step) directly to the mounted *backup* volume.

IMPORTANT: If you store the backup volume in any location that an adversary can repeatedly access (for example, on a device kept in a bank's safe deposit box), you should repeat all of the above steps (including the step 2) each time you want to back up the volume (see below).

If you follow the above steps, you will help prevent adversaries from finding out:

- Which sectors of the volumes are changing (because you always follow step 2). This is particularly important, for example, if you store the backup volume on a device kept in a bank's safe deposit box (or in any other location that an adversary can repeatedly access) and the volume contains a hidden volume (for more information, see the subsection *Security Precautions Pertaining to Hidden Volumes* in the chapter *Plausible Deniability*).

- That one of the volumes is a backup of the other

General Notes

If you store the backup volume in any location where an adversary can make a copy of the volume, consider encrypting the volume with a cascade of ciphers (for example, with AES-Twofish-Serpent). Otherwise, if the volume is encrypted only with a single encryption algorithm and the algorithm is later broken (for example, due to advances in cryptanalysis), the attacker might be able to decrypt his copies of the volume. The probability that three distinct encryption algorithms will be broken is significantly lower than the probability that only one of them will be broken.

Troubleshooting

This section presents possible solutions to common problems that you may run into when using TrueCrypt. If your problem is not listed here, it might be listed in one of the following sections:

Incompatibilities
Known Issues & Limitations
Frequently Asked Questions

PROBLEM:

Writing/reading to/from volume is very slow even though, according to the benchmark, the speed of the cipher that I'm using is higher than the speed of the hard drive.

PROBABLE CAUSE:

This is probably caused by an interfering application.

POSSIBLE SOLUTION:

First, make sure that your TrueCrypt container does not have a file extension that is reserved for executable files (for example, .exe, .sys, or .dll). If it does, Windows and antivirus software may interfere with the container and adversely affect the performance of the volume.

Second, disable or uninstall any application that might be interfering, which usually is antivirus software or automatic disk defragmentation tool, etc. In case of antivirus software, it often helps to turn off real-time (on-access) scanning in the preferences of the antivirus software. If it does not help, try temporarily disabling the virus protection software. If this does not help either, try uninstalling it completely and restarting your computer subsequently.

PROBLEM:

*After successfully mounting a volume, Windows reports "*This device does not contain a valid file system*" or a similar error.*

PROBABLE CAUSE:

The file system on the TrueCrypt volume may be corrupted (or the volume is unformatted).

POSSIBLE SOLUTION:

You can use filesystem repair tools supplied with your operating system to attempt to repair the filesystem on the TrueCrypt volume. In Windows, it is the '*chkdsk*' tool. TrueCrypt provides an easy way to use this tool on a TrueCrypt volume: First, make a backup copy of the TrueCrypt volume (because the '*chkdsk*' tool might damage the filesystem even more) and then mount it. Right-click the mounted volume in the main TrueCrypt window (in the drive list) and from the context menu select '*Repair Filesystem*'.

PROBLEM:

When trying to create a hidden volume, its maximum possible size is unexpectedly small (there is much more free space than this on the outer volume).

PROBABLE CAUSE:

Fragmentation

OR

Too small cluster size + too many files/folders in the root directory of the outer volume.

POSSIBLE SOLUTION:

Note: The following solution applies only to hidden volumes created within FAT volumes.

Defragment the outer volume (mount it, right-click its drive letter in the '*Computer*' or '*My Computer*' window, click *Properties*, select the *Tools* tab, and click '*Defragment Now*'). After the volume is defragmented, exit *Disk Defragmenter* and try to create the hidden volume again.

If this does not help, delete *all* files and folders on the outer volume by pressing Shift+Delete, not by formatting, (do not forget to disable the Recycle Bin and System Restore for this drive beforehand) and try creating the hidden volume on this *completely empty* outer volume again (for testing purposes only). If the maximum possible size of the hidden volume does not change even now, the cause of the problem is very likely an extended root directory. If you did not use the '*Default*' cluster size (the last step in the Wizard), reformat the outer volume and this time leave the cluster size at '*Default*'.

If it does not help, reformat the outer volume again and copy less files/folders to its root folder than you did last time. If it does not help, keep reformatting and decreasing the number of files/folders in the root folder. If this is unacceptable or if it does not help, reformat the outer volume and select a larger cluster size. If it does not help, keep reformatting and increasing the cluster size, until the problem is solved. Alternatively, try creating a hidden volume within an NTFS volume.

PROBLEM:

I cannot encrypt a partition/device because TrueCrypt Volume Creation Wizard says it is in use.

POSSIBLE SOLUTION:

Close, disable, or uninstall all programs that might be using the partition/device in any way (for example an anti-virus utility). If it does not help, right-click the '*Computer*' (or '*My Computer*') icon on your desktop and select *Manage -> Storage -> Disk Management.* Then right-click the partition that you want to encrypt, and click *Change Drive Letter and Paths.* Then click *Remove* and *OK.* Restart the operating system.

PROBLEM:

When creating a hidden volume, the Wizard reports that the outer volume cannot be locked.

PROBABLE CAUSE:

The outer volume contains files being used by one or more applications.

POSSIBLE SOLUTION:

Close all applications that are using files on the outer volume. If it does not help, try disabling or uninstalling any anti-virus utility you use and restarting the system subsequently.

PROBLEM:

One of the following problems occurs:

A TrueCrypt volume cannot be mounted
NTFS TrueCrypt volumes cannot be created

In addition, the following error may be reported: "*The process cannot access the file because it is being used by another process.*"

PROBABLE CAUSE:

This is probably caused by an interfering application. Note that this is not a bug in TrueCrypt. The operating system reports to TrueCrypt that the device is locked for an exclusive access by an application (so TrueCrypt is not allowed to access it).

POSSIBLE SOLUTION:

It usually helps to disable or uninstall the interfering application, which is usually an anti-virus utility, a disk management application, etc.

PROBLEM:

When accessing a file-hosted container shared over a network, “insufficient memory” or "not enough server storage is available" error is reported.

PROBABLE CAUSE:

IRPStackSize in the Windows registry may have been set to a too small value.

POSSIBLE SOLUTION:

Locate the *IRPStackSize* key in the Windows registry and set it to a higher value. Then restart the system. If the key does not exist in your Windows registry, create it at:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
and set its value to 16 or higher. Then restart the system. For more information, see:
http://support.microsoft.com/kb/285089/ and http://support.microsoft.com/kb/177078/

Incompatibilities

Activation of Adobe Photoshop® and Other Products Using FLEXnet Publisher® / SafeCast

*Note: The issue described below does **not** affect you if you use TrueCrypt 5.1 or later and a non-cascade encryption algorithm (i.e., AES, Serpent, or Twofish).[*] The issue also does **not** affect you if you do not use pre-boot authentication (see the chapter* System Encryption*).*

Acresso FLEXnet Publisher activation software, formerly Macrovision SafeCast, (used for activation of third-party software, such as *Adobe Photoshop*) writes data to the first drive track. If this happens when your system partition/drive is encrypted by TrueCrypt, a portion of the TrueCrypt Boot Loader will be damaged and you will not be able to start Windows. In that case, please use your TrueCrypt Rescue Disk to regain access to your system. There are two ways to do so:

1. You may keep the third-party software activated but you will need to boot your system from the TrueCrypt Rescue Disk CD/DVD *every time*. Just insert your Rescue Disk into your CD/DVD drive and then enter your password in the Rescue Disk screen.

2. If you do not want to boot your system from the TrueCrypt Rescue Disk CD/DVD every time, you can restore the TrueCrypt Boot Loader on the system drive. To do so, in the Rescue Disk screen, select *Repair Options* > *Restore TrueCrypt Boot Loader*. However, note that this will deactivate the third-party software.

For information on how to use your TrueCrypt Rescue Disk, please see the chapter *TrueCrypt Rescue Disk*.

Possible permanent solution: Upgrade to TrueCrypt 5.1 or later, decrypt the system partition/drive, and then re-encrypt it using a non-cascade encryption algorithm (i.e., AES, Serpent, or Twofish).*

Please note that this not a bug in TrueCrypt (the issue is caused by inappropriate design of SafeCast).

[*] The reason is that the TrueCrypt Boot Loader is smaller than the one used for cascades of ciphers and, therefore, there is enough space in the first drive track for a backup of the TrueCrypt Boot Loader. Hence, whenever the TrueCrypt Boot Loader is damaged, its backup copy is run automatically instead.

Known Issues & Limitations

Known Issues

> **It is strongly recommended that you also read the latest online version of the list of known issues at:** http://www.truecrypt.org/docs/?s=issues-and-limitations

- (There were no confirmed issues when this document was created.)

Limitations

- On Windows XP/2003, TrueCrypt does not support encrypting an entire system drive that contains extended (logical) partitions. You can encrypt an entire system drive provided that it contains only primary partitions. Extended (logical) partitions must not be created on any system drive that is partially or fully encrypted (only primary partitions may be created on it). *Note*: If you need to encrypt an entire drive containing extended partitions, you can encrypt the system partition and, in addition, create partition-hosted TrueCrypt volumes within any non-system partitions on the drive. Alternatively, you may want to consider upgrading to Windows Vista or a later version of Windows.

- TrueCrypt currently does not support encrypting a system drive that has been converted to a dynamic disk.

- TrueCrypt volume passwords must consist only of printable ASCII characters. Non-ASCII characters in passwords are not supported and may cause various problems (e.g., inability to mount a volume).

- Due to a Windows 2000 issue, TrueCrypt does not support the Windows Mount Manager under Windows 2000. Therefore, some Windows 2000 built-in tools, such as Disk Defragmenter, do not work on TrueCrypt volumes. Furthermore, it is not possible to use the Mount Manager services under Windows 2000, e.g., assign a mount point to a TrueCrypt volume (i.e., attach a TrueCrypt volume to a folder).

- The Windows Volume Shadow Copy Service is currently supported only for partitions within the key scope of system encryption (for example, a system partition encrypted by TrueCrypt or a non-system partition located on a system drive encrypted by TrueCrypt). Note: For other types of volumes, the Volume Shadow Copy Service is not supported because the documentation for the necessary API is available from Microsoft only under a non-disclosure agreement.

- Encrypted partitions cannot be resized.

- When the system partition/drive is encrypted, the system cannot be upgraded (for example, from Windows XP to Windows Vista) or repaired from within the pre-boot environment (using a Windows setup CD/DVD). In such cases, the system partition/drive must be decrypted first.

Note: The system can be *updated* (security patches, service packs, etc.) without any problems even when the system partition/drive is encrypted.

- When the notebook battery power is low, Windows may omit sending the appropriate messages to running applications when the computer is entering power saving mode. Therefore, TrueCrypt may fail to auto-dismount volumes in such cases.

- TrueCrypt-encrypted floppy disks: When a floppy disk is ejected and another one is inserted, data read/written from/to the disk will be corrupted. Note that this affects *only raw* floppy disk volumes (not file-hosted TrueCrypt containers stored on floppy disks).

Frequently Asked Questions

The latest version of the TrueCrypt FAQ is available at: http://www.truecrypt.org/faq.php

Q: *Is there a "Quick Start Guide" or some tutorial for beginners?*

A: *Yes. The first chapter,* Beginner's Tutorial*, contains screenshots and step-by-step instructions on how to create, mount, and use a TrueCrypt volume.*

Q: *Can TrueCrypt encrypt a partition/drive where Windows is installed?*

A: *Yes (see the chapter* System Encryption*).*

Q: *Can I directly play a video (.avi, .mpg, etc.) stored on a TrueCrypt volume?*

A: *Yes, TrueCrypt-encrypted volumes are like normal disks. You provide the correct password (and/or keyfile) and mount (open) the TrueCrypt volume. When you double click the icon of the video file, the operating system launches the application associated with the file type – typically a media player. The media player then begins loading a small initial portion of the video file from the TrueCrypt-encrypted volume to RAM (memory) in order to play it. While the portion is being loaded, TrueCrypt is automatically decrypting it (in RAM). The decrypted portion of the video (stored in RAM) is then played by the media player. While this portion is being played, the media player begins loading next small portion of the video file from the TrueCrypt-encrypted volume to RAM (memory) and the process repeats.*

The same goes for video recording: Before a chunk of a video file is written to a TrueCrypt volume, TrueCrypt encrypts it in RAM and then writes it to the disk. This process is called on-the-fly encryption/decryption and it works for all file types, not only for video files.

Q: *Will TrueCrypt be open-source and free forever?*

A: *Yes, it will. We will never create a commercial version of TrueCrypt, as we believe in open-source and free security software.*

Q: *Is it possible to donate to the TrueCrypt project?*

A: *Yes. For more information, please visit http://www.truecrypt.org/donations/*

Q: *I forgot my password – is there any way to recover the files from my TrueCrypt volume?*

A: TrueCrypt does not contain any mechanism or facility that would allow partial or complete recovery of your encrypted data without knowing the correct password or the key used to encrypt the data. The only way to recover your files is to try to "crack" the password or the key, but it could take thousands or millions of years depending on the length and quality of the password/keyfiles, on software/hardware efficiency, and other factors.

Q: *Does TrueCrypt also encrypt file names and folder names?*

A: *Yes. The entire file system within a TrueCrypt volume is encrypted (including file names, folder names, and contents of every file). This applies to both types of TrueCrypt volumes – i.e., to file containers (virtual TrueCrypt disks) and to TrueCrypt-encrypted partitions/devices.*

Q: *How can I use TrueCrypt on a USB flash drive?*

A: You have two options:

1) *Encrypt the entire USB flash drive. However, you will not be able run TrueCrypt from the USB flash drive.*
   Note: Windows does not support multiple partitions on USB flash drives.

2) *Create a TrueCrypt file container on the USB flash drive (for information on how to do so, see the chapter* Beginner's Tutorial*). If you leave enough space on the USB flash drive (choose an appropriate size for the TrueCrypt container), you will also be able to store TrueCrypt on the USB flash drive (along with the container – not* in *the container) and you will be able to run TrueCrypt from the USB flash drive (see also the chapter* Traveler Mode*).*

Q: *Is it possible to boot Windows installed in a hidden TrueCrypt volume?*

A: *Yes, it is (as of TrueCrypt 6.0). For more information, please see the section* Hidden Operating System*.*

Q: *Will I be able to mount my TrueCrypt volume (container) on any computer?*

A: *Yes, TrueCrypt volumes (in contrast to TrueCrypt-encrypted physical system partitions/drives) are independent of the operating system. You will be able to mount your TrueCrypt volume on any computer on which you can run TrueCrypt (see also the question "*Can I use TrueCrypt in Windows if I do not have administrator privileges?*").*

Q: *Can I unplug or turn off a hot-plug device (for example, a USB flash drive or USB hard drive) when there is a mounted TrueCrypt volume on it?*

A: *Before you unplug or turn off the device, you should always dismount the TrueCrypt volume in TrueCrypt first, and then perform the '*Eject*' operation if available (right-click the device in the '*Computer*' or '*My Computer*' list), or use the '*Safely Remove Hardware*' function (built in Windows, accessible via the taskbar notification area). Otherwise, data loss may occur.*

Q: *What is a hidden operating system?*

See the section *Hidden Operating System*.

Q: *Will I be able to mount my TrueCrypt partition/container after I reinstall or upgrade the operating system?*

A: *Yes, TrueCrypt volumes are independent of the operating system. However, you need to make sure your operating system installer does not format the partition where your TrueCrypt volume resides.*

Note: If the system partition/drive is encrypted and you want to reinstall or upgrade Windows, you need to decrypt it first (select System > Permanently Decrypt System Partition/Drive).

Q: *Can I upgrade from an older version of TrueCrypt to the latest version without any problems?*

A: *Generally, yes. However, before upgrading, please read the release notes for all versions of TrueCrypt that have been released since your version was released. If there are any known issues or incompatibilities related to upgrading from your version to a newer one, they will be listed in the release notes.*

Q: *Can I upgrade TrueCrypt if the system partition/drive is encrypted or do I have to decrypt it first?*

A: *Yes, generally, you can upgrade to the latest version without decrypting the system partition/drive (just run the TrueCrypt installer and it will automatically upgrade TrueCrypt on the system). However, before upgrading, please read the release notes for all versions of TrueCrypt that have been released since your version was released. If there are any known issues or incompatibilities related to upgrading from your version to a newer one, they will be listed in the release notes. Note: You cannot downgrade TrueCrypt if the system partition/drive is encrypted.*

Q: *I use pre-boot authentication. Can I prevent a person (adversary) that is watching me start my computer from knowing that I use TrueCrypt?*

A: *Yes (as of TrueCrypt 6.1). To do so, boot the encrypted system, start TrueCrypt, select* Settings > System Encryption*, enable the option '*Do not show any texts in the pre-boot authentication screen*' and click* OK*. Then, when you start the computer, no texts will be displayed by the TrueCrypt boot loader (not even when you enter the wrong password). The computer will appear to be "frozen" while you can type your password. It is, however, important to note that if the adversary can analyze the content of the hard drive, he can still find out that it contains the TrueCrypt boot loader.*

Q: *I use pre-boot authentication. Can I configure the TrueCrypt Boot Loader to display only a fake error message?*

A: *Yes (as of TrueCrypt 6.1). To do so, boot the encrypted system, start TrueCrypt, select* Settings > System Encryption*, enable the option '*Do not show any texts in the pre-boot authentication screen*' and enter the fake error message in the corresponding field (for example, the "*Missing operating system*" message, which is normally displayed by the Windows boot loader if it finds no Windows boot partition). It is, however, important to note that if the adversary can analyze the content of the hard drive, he can still find out that it contains the TrueCrypt boot loader.*

Q: *How do I mount a hidden volume?*

A: *A hidden volume can be mounted the same way as a standard TrueCrypt volume: Click '*Select File*' or '*Select Device*' to select the outer/host volume (important: make sure the volume is not mounted). Then click Mount, and enter the password for the hidden volume. Whether the hidden or the outer volume will be mounted is determined by the entered password (i.e., when you enter the password for the outer volume, then the outer volume will be mounted; when you enter the password for the hidden volume, the hidden volume will be mounted).*

Note: TrueCrypt first attempts to decrypt the standard volume header using the entered password. If it fails, it loads the area of the volume where a hidden volume header can be stored (i.e. the bytes 65536–131071, which contain solely random data when there is no hidden volume within the volume) to RAM and attempts to decrypt it using the entered password. Note that hidden volume headers cannot be identified, as they appear to consist entirely of random data. If the header is successfully decrypted (for information on how TrueCrypt determines that it was successfully decrypted, see the section *Encryption Scheme*), the information about the size of the hidden volume is retrieved from the decrypted header (which is still stored in RAM), and the hidden volume is mounted (its size also determines its offset).

Further information may be found in the section Hidden Volume*.*

Q: *Can I save data to the decoy system partition without risking damage to the hidden system partition?*

A: *Yes. You can write data to the decoy system partition anytime without any risk that the hidden volume will get damaged (because the decoy system is* not *installed within the same partition as the hidden system). For more information, see the section* Hidden Operating System*.*

Q: *Can I use TrueCrypt in Windows if I do not have administrator privileges?*

See the chapter '*Using TrueCrypt Without Administrator Privileges*'.

Q: *Does TrueCrypt save my password to a disk?*

A: *No.*

Q: *Is some hash of my password stored somewhere?*

A: *No.*

Q: *Is it possible to install an application to a TrueCrypt volume and run it from there?*

A: *Yes.*

Q*: How does TrueCrypt verify that the correct password was entered?*

See the chapter *Technical Details*, section *Encryption Scheme*.

Q: *Does TrueCrypt support hardware/software RAID and Windows dynamic volumes?*

A: *Yes. However, if you use Windows XP/2000/2003, please read the following notes on dynamic volumes (the notes do not apply to Windows Vista and later). If you intend to format a Windows dynamic volume as a TrueCrypt volume, keep in mind that after you create the Windows dynamic volume (using the* Windows Disk Management *tool), you must restart the operating system in order for the volume to be available/displayed in the 'Select Device' dialog window of the TrueCrypt Volume Creation Wizard. Also note that, in the 'Select Device' dialog window, a Windows dynamic volume is not displayed as a single device (item). Instead, all volumes that the Windows dynamic volume consists of are displayed and you can select* any *of them in order to format the entire Windows dynamic volume.*

Q: *Is it possible to mount a TrueCrypt container that is stored on a CD or DVD?*

A: *Yes. However, if you need to mount a TrueCrypt volume that is stored on a read-only medium (such as a CD or DVD) under Windows 2000, the file system within the TrueCrypt volume must be FAT (Windows 2000 cannot mount an NTFS file system on read-only media).*

Q: *Can I run TrueCrypt if I don't install it?*

A: *Yes, see the chapter* Traveler Mode*.*

Q: *Why does Windows Vista ask me for permission to run TrueCrypt every time I run it in 'traveler' mode?*

A: *When you run TrueCrypt in traveler mode, TrueCrypt needs to load and start the TrueCrypt device driver. TrueCrypt needs a device driver to provide transparent on-the-fly encryption/decryption, and users without administrator privileges cannot start device drivers in Windows. Therefore, Windows Vista asks you for permission to run TrueCrypt with administrator privileges.*

Note that if you install TrueCrypt on the system (as opposed to running TrueCrypt in traveler mode), you will not be asked for permission every time you run it.

Q: *What is the maximum possible size of a TrueCrypt volume?*

A: *The maximum possible size of a TrueCrypt volume is 8589934592 GB. However, due to security reasons, the maximum allowed volume size is 1 PB (1,048,576 GB), as the amount of data that is considered secure to be encrypted using a single key depends, among other factors, on the block size of the encryption algorithm. In addition, you need to take into account other limiting factors. For instance, file system constraints, limitations of the hardware connection standard and of the operating system, etc.*

Q: *Do I have to dismount TrueCrypt volumes before shutting down or restarting Windows?*

A: *No. TrueCrypt automatically dismounts all mounted TrueCrypt volumes on system shutdown/restart.*

Q: *Which type of TrueCrypt volume is better – partition or file container?*

A: *File containers are normal files so you can work with them as with any normal files (file containers can be, for example, moved, renamed, and deleted the same way as normal files). Partitions/drives may be better as regards performance. Note that reading and writing to/from a file container may take significantly longer when the container is heavily fragmented. To solve this problem, defragment the file system in which the container is stored (when the TrueCrypt volume is dismounted).*

Q: *What's the recommended way to backup a TrueCrypt volume?*

See the chapter *How to Back Up Securely.*

Q: *What will happen if I format a TrueCrypt partition?*

See the question "*Is it possible to change the file system of an encrypted volume?*" in this FAQ.

Q: *Is it possible to change the file system of an encrypted volume?*

A: *Yes, when mounted, TrueCrypt volumes can be formatted as FAT12, FAT16, FAT32, NTFS, or any other file system. TrueCrypt volumes behave as standard disk devices so you can right-click the device icon (for example in the '*Computer*' or '*My Computer*' list) and select 'Format'. The actual volume contents will be lost. However, the whole volume will remain encrypted. If you format a TrueCrypt-encrypted partition when the TrueCrypt volume that the partition hosts is not mounted, then the volume will be destroyed, and the partition will not be encrypted anymore (it will be empty).*

Q: *Can I configure TrueCrypt to start, prompt me for password(s), and mount my volume(s) automatically whenever Windows starts?*

A: Yes. To do so, follow these steps:

1. *Mount the volume(s) and then select '*Volumes*' -> '*Save Currently Mounted Volumes as Favorite*'.*
2. *Select '*Settings*' -> '*Preferences*'. In the '*Preferences*' window in the section '*Actions to perform upon log on to Windows*', enable the option '*Mount favorite volumes'.
3. *In the '*Preferences*' window, click '*OK*'.*

*Alternatively, if the volume(s) is/are partition/device-hosted and if you do not need to mount it/them to particular drive letter(s) every time, you may skip step 1 and in the '*Preferences*' window in the section '*Actions to perform upon log on to Windows*' enable the option '*Mount all devices-hosted TrueCrypt volumes*' (instead of '*Mount favorite volumes*').*

Q: *Is it possible to change the password for a hidden volume?*

A: *Yes, the password change dialog works both for standard and hidden volumes. Just type the password for the hidden volume in the 'Current Password' field of the 'Volume Password Change' dialog.*

Remark: TrueCrypt first attempts to decrypt the standard volume header and if it fails, it attempts to decrypt the area within the volume where the hidden volume header may be stored (if there is a hidden volume within). In case it is successful, the password change applies to the hidden volume. (Both attempts use the password typed in the '*Current Password*' field.)

Q: *When I use HMAC-RIPEMD-160, is the size of the header encryption key only 160 bits?*

A: No, TrueCrypt never uses an output of a hash function (nor of a HMAC algorithm) directly as an encryption key. See the section 'Header Key Derivation, Salt, and Iteration Count' *for more information.*

Q: *Can I change the header key derivation algorithm (for example, from HMAC-RIPEMD-160 to HMAC-SHA-512) without losing data stored on the volume?*

A: *Yes. To do so, select* Volumes -> Set Header Key Derivation Algorithm*.*

Q: *How do I burn a TrueCrypt container larger than 2 GB onto a DVD?*

A: *The DVD burning software you use should allow you to select the format of the DVD. If it does, select the UDF format (ISO format does not support files larger than 2 GB).*

Q: *What license is TrueCrypt distributed under?*

A: *The text of the license is contained in the file* License.txt *that is included in the TrueCrypt binary and source code distribution packages, and is also available at* *http://www.truecrypt.org/legal/license*.

***Q:* The Windows file selector remembers the path of the last container I mount or the path of the last selected keyfile. Is there a way to prevent this?**

A: *Yes, there is. If you have not done so yet, upgrade to TrueCrypt 4.2a or later. Run TrueCrypt and make sure the option 'Never save history' in the main window is enabled. If you do not want to enable the option 'Never save history', you can avoid using the Windows file selector by dragging the icon of the container onto the 'TrueCrypt.exe' icon (TrueCrypt will be automatically launched then), or dragging it onto the TrueCrypt program window. Likewise, a keyfile can be selected by dragging its icon onto the Keyfiles window or onto the password entry window.*

Q: Can I encrypt a partition/drive without losing the data currently stored on it?

A: *Yes, but the following conditions must be met: If you want to encrypt an entire system drive (which may contain multiple partitions) or a system partition (in other words, if you want to encrypt a drive or partition where Windows is installed) you can do so provided that you use TrueCrypt 5.0*

or later and that you use Windows XP or a later version of Windows (such as Windows Vista). If you want to encrypt a non-system partition in place, you can do so provided that it contains an NTFS filesystem, that you use TrueCrypt 6.1 or later, and that you use Windows Vista or a later version of Windows (such as Windows 2008).

Q: *Can I use tools like* chkdsk*, Disk Defragmenter, etc. on the contents of a mounted TrueCrypt volume?*

A: *Yes, TrueCrypt volumes behave like real physical disk devices, so it is possible to use any filesystem checking/repairing/defragmenting tools on the contents of a mounted TrueCrypt volume.*

Q: *Is it possible to use TrueCrypt without leaving any 'traces' on unencrypted Windows?*

A: *Yes. This can be achieved by running TrueCrypt in traveler mode under BartPE. BartPE stands for "Bart's Preinstalled Environment", which is essentially the Windows operating system prepared in a way that it can be entirely stored on and booted from a CD/DVD (registry, temporary files, etc., are stored in RAM – hard drive is not used at all and does not even have to be present). The freeware Bart's PE Builder can transform a Windows XP installation CD into BartPE. As of TrueCrypt 3.1, you do not need any TrueCrypt plug-in for BartPE. Just boot BartPE, download the TrueCrypt self-extracting package to the RAM disk (which BartPE creates), run it, extract its content to the RAM disk, and then run the file 'TrueCrypt.exe' from the RAM disk.*

Note: You may also want to consider creating a hidden operating system (for more information, see the section Hidden Operating System*).*

Q: *Can I mount a TrueCrypt volume stored in another TrueCrypt volume?*

A: *Yes, TrueCrypt volumes can be nested without any limitation.*

Q: *Can I run TrueCrypt with another on-the-fly disk encryption tool on one system?*

A: *We are not aware of any on-the-fly encryption tool that would cause problems when run with TrueCrypt, or vice versa.*

Q: *Can I resize a TrueCrypt partition?*

A: *Unfortunately, TrueCrypt does not support this. Resizing a TrueCrypt partition using a program such as PartitionMagic will, in most cases, corrupt its contents.*

Q: *Does TrueCrypt run on Windows Vista x64 (64-bit) Edition?*

A: *Yes. Note: All .sys and .exe files of TrueCrypt are digitally signed with the digital certificate of the TrueCrypt Foundation, which was issued by the certification authority GlobalSign.*

Q: *Does TrueCrypt run on Mac OS X?*

A: *Yes.*

Q: *Does TrueCrypt run on Linux?*

A: *Yes.*

Q: *Can I mount my TrueCrypt volume under Windows, Mac OS X, and Linux?*

A: *Yes, TrueCrypt volumes are fully cross-platform.*

Q: *What will happen when a part of a TrueCrypt volume becomes corrupted?*

A: *In encrypted data, one corrupted bit usually corrupts the whole ciphertext block in which it occurred. The ciphertext block size used by TrueCrypt is 16 bytes (i.e., 128 bits). The mode of operation used by TrueCrypt ensures that if data corruption occurs within a block, the remaining blocks are not affected (for more information, see the section Modes of Operation). See also the question '*What do I do when the encrypted filesystem on my TrueCrypt volume is corrupted?*'*

Q: *What do I do when the encrypted filesystem on my TrueCrypt volume is corrupted?*

A: *File system within a TrueCrypt volume may become corrupted in the same way as any normal unencrypted file system. When that happens, you can use filesystem repair tools supplied with your operating system to fix it. In Windows, it is the '*chkdsk*' tool. TrueCrypt provides an easy way to use this tool on a TrueCrypt volume: First, make a backup copy of the TrueCrypt volume (because the '*chkdsk*' tool might damage the filesystem even more) and then mount it. Right-click the mounted volume in the main TrueCrypt window (in the drive list) and from the context menu select* 'Repair Filesystem'.

Q: *We use TrueCrypt in a corporate/enterprise environment. Is there a way for an administrator to reset a volume password or pre-boot authentication password when a user forgets it (or loses a keyfile)?*

A: *Yes. Note that there is no "back door" implemented in TrueCrypt. However, there* is *a way to "reset" volume passwords/keyfiles and pre-boot authentication passwords. After you create a volume, back up its header to a file (select* Tools -> Backup Volume Header*) before you allow a non-admin user to use the volume. Note that the volume header (which is encrypted with a header key derived from a password/keyfile) contains the master key with which the volume is encrypted. Then ask the user to choose a password, and set it for him/her (*Volumes -> Change Volume Password*); or generate a user keyfile for him/her. Then you can allow the user to use the volume and to change the password/keyfiles without your assistance/permission. In case he/she forgets his/her password or loses his/her keyfile, you can "reset" the volume password/keyfiles to your original admin password/keyfiles by restoring the volume header from the backup file (*Tools -> Restore Volume Header*).*

Similarly, you can reset a pre-boot authentication password. To create a backup of the master key data (that will be stored on a TrueCrypt Rescue Disk and encrypted with your administrator

*password), select '*System*' > '*Create Rescue Disk*'. To set a user pre-boot authentication password, select '*System*' > '*Change Password*'. To restore your administrator password, boot the TrueCrypt Rescue Disk, select '*Repair Options*' > '*Restore key data*', and enter your administrator password. Note: It is not required to burn each TrueCrypt Rescue Disk ISO image to a CD/DVD. You can maintain a central repository of ISO images for all workstations (rather than a repository of CDs/DVDs). For more information, see the section* Command Line Usage *(option* /noisocheck*).*

Q: *It is possible to access a single TrueCrypt volume simultaneously from multiple operating systems (for example, a volume shared over a network)?*

Please see the chapter Sharing over Network*.*

Q: Can a user access his or her TrueCrypt volume via a network?

Please see the chapter Sharing over Network*.*

Q: *I encrypted a non-system partition, but its original drive letter is still visible in the 'My Computer' list. When I double click this drive letter, Windows asks if I want to format the drive. Is there a way to hide or free this drive letter?*

A: *Yes, to free the drive letter follow these steps:*

1. *Right-click the '*Computer*' (or '*My Computer*') icon on your desktop or in the Start Menu and select* Manage*. The '*Computer Management*' window should appear.*
2. *From the list on the left, select '*Disk Management*' (within the* Storage *sub-tree).*
3. *Right-click the encrypted partition and select '*Change Drive Letter and Paths*'.*
4. *Click* Remove*.*
5. If Windows prompts you to confirm the action, click Yes.

Q: *How do I remove or undo encryption if I do not need it anymore? How do I permanently decrypt a volume?*

Please see the chapter How to Remove Encryption*.*

Q: *What will change when I enable the option 'Mount volumes as removable media'?*

A: *You can enable this option, for example, to prevent Windows from automatically creating the '*Recycled*' and/or the '*System Volume Information*' folders on TrueCrypt volumes (in Windows, these folders are used by the Recycle Bin and System Restore facilities). However, there are some disadvantages. For example, when you enable this option, the '*Computer*' (or '*My Computer*') list will not show free space on the volume (note that this is a Windows limitation, not a bug in TrueCrypt).*

Q: *Is it secure to create a new container by cloning an existing container?*

A: *You should always use the Volume Creation Wizard to create a new TrueCrypt volume. If you copy a container and then start using both this container and its clone in a way that both eventually contain different data, then you might aid cryptanalysis (both volumes would share a single key set). See also the chapter* How to Back Up Securely*.*

Q: Do I have to “wipe” free space and/or files on a TrueCrypt volume?

Remark: to "wipe" = to securely erase; to overwrite sensitive data in order to render them unrecoverable.

A: *If you believe that an adversary will be able to decrypt the volume (for example that he will make you reveal the password), then the answer is yes. Otherwise, it is not necessary, because the volume is entirely encrypted.*

Q: *How does TrueCrypt know which encryption algorithm my TrueCrypt volume has been encrypted with?*

Please see the section Encryption Scheme *(chapter* Technical Details*).*

How to Remove Encryption

Please note that TrueCrypt can in-place decrypt only **system partitions and system drives** (select *System* > *Permanently Decrypt System Partition/Drive*). If you need to remove encryption (e.g., if you no longer need encryption) from a **non-system volume**, please follow these steps:

1. Mount the TrueCrypt volume.

2. Move all files from the TrueCrypt volume to any location outside the TrueCrypt volume (note that the files will be decrypted on-the-fly).

3. Dismount the TrueCrypt volume.

4. **If the TrueCrypt volume is file-hosted**, delete it (the container) just like you delete any other file.

   If the volume is partition-hosted (applies also to USB flash drives), in addition to the steps 1-3, do the following:

   a. Right-click the '*Computer*' (or '*My Computer*') icon on your desktop, or in the Start Menu, and select *Manage*. The '*Computer Management*' window should appear.
   b. In the *Computer Management* window, from the list on the left, select '*Disk Management*' (within the *Storage* sub-tree).
   c. Right-click the partition you want to decrypt and select '*Change Drive Letter and Paths*'.
   d. The '*Change Drive Letter and Paths*' window should appear. If no drive letter is displayed in the window, click *Add*. Otherwise, click *Cancel*.
   If you clicked *Add*, then in the '*Add Drive Letter or Path*' (which should have appeared), select a drive letter you want to assign to the partition and click *OK*.
   e. In the *Computer Management* window, right-click the partition you want to decrypt again and select *Format*. The *Format* window should appear.
   f. In the *Format* window, click *OK*. After the partition is formatted, it will no longer be required to mount it with TrueCrypt to be able to save or load files to/from the partition.

   If the volume is device-hosted (i.e., there are no partitions on the device, and the device is entirely encrypted), in addition to the steps 1-3, do the following:

   a. Right-click the '*Computer*' (or '*My Computer*') icon on your desktop, or in the Start Menu, and select *Manage*. The '*Computer Management*' window should appear.
   b. In the *Computer Management* window, from the list on the left, select '*Disk Management*' (within the *Storage* sub-tree).
   c. Right-click the area representing the storage space of the encrypted device and select '*New Partition*' or '*New Simple Volume*'.
   d. WARNING: Before you continue, make sure you have selected the correct device, as all files stored on it will be lost. The '*New Partition Wizard*' or '*New Simple Volume Wizard*' window should appear now; follow its instructions to create a new partition on the device. After the partition is created, it will no longer be required to mount the device with TrueCrypt to be able to save or load files to/from the device.

Uninstalling TrueCrypt

To uninstall TrueCrypt on Windows XP, select *Start* menu > *Settings* > *Control Panel* > *Add or Remove Programs* > *TrueCrypt* > *Change/Remove*. To uninstall TrueCrypt on Windows Vista, select *Start* menu > *Control Panel* > *Programs - Uninstall a program* > *TrueCrypt* > *Change/Remove*.

No TrueCrypt volume will be removed when you uninstall TrueCrypt. You will be able to mount your TrueCrypt volume(s) again after you install TrueCrypt or when you run it in 'traveler' mode.

TrueCrypt System Files & Application Data

Note: `%windir%` is the main Windows installation path (e.g., `C:\WINDOWS`)

The TrueCrypt driver:

`%windir%\SYSTEM32\DRIVERS\truecrypt.sys` (32-bit Windows)
or:
`%windir%\SysWOW64\drivers\truecrypt.sys` (64-bit Windows)

Note: This file is not present if TrueCrypt is run in 'traveler' mode.

TrueCrypt settings / application data:

The following files are saved in the folder where application data are normally saved on your system (for example, in `C:\Documents and Settings\`*`UserName`*`\Application Data\TrueCrypt\`, where *`UserName`* is your Windows user name). In traveler mode, these files are saved to the folder from which you run the file *TrueCrypt.exe* (i.e., the folder in which *TrueCrypt.exe* resides). **WARNING: Note that TrueCrypt does *not* encrypt those files (unless TrueCrypt encrypts the system partition/drive).**

`Configuration.xml`

`Original System Loader` (a backup of the original content of the first drive track made before the TrueCrypt Boot Loader was written to it).
Note: This file is absent if the system partition/drive has not been encrypted.

`System Encryption.xml` (temporary configuration file used during the process of encryption/decryption of the system partition/drive).

`Default Keyfiles.xml`
Note: This file may be absent if the corresponding TrueCrypt feature is not used.

`Favorite Volumes.xml`
Note: This file may be absent if the corresponding TrueCrypt feature is not used.

`History.xml` (the list of last twenty files/devices attempted to be mounted as TrueCrypt volumes or attempted to be used as hosts for TrueCrypt volumes; this feature can be disabled – for more information, see the section *Never Save History*)
Note: This file may be absent if the corresponding TrueCrypt feature is not used.

Technical Details

Notation

C	Ciphertext block
$D_K()$	Decryption algorithm using encryption/decryption key K
$E_K()$	Encryption algorithm using encryption/decryption key K
$H()$	Hash function
i	Block index for n-bit blocks; n is context-dependent
K	Cryptographic key
P	Plaintext block
^	Bitwise exclusive-OR operation (XOR)
$\oplus$	Modulo 2^n addition, where n is the bit size of the left-most operand and of the resultant value (e.g., if the left operand is a 1-bit value, and the right operand is a 2-bit value, then: $1 \oplus 0 = 1$; $1 \oplus 1 = 0$; $1 \oplus 2 = 1$; $1 \oplus 3 = 0$; $0 \oplus 0 = 0$; $0 \oplus 1 = 1$; $0 \oplus 2 = 0$; $0 \oplus 3 = 1$)
$\otimes$	Modular multiplication of two polynomials over the binary field GF(2) modulo $x^{128}+x^7+x^2+x+1$ (GF stands for Galois Field)
||	Concatenation

Encryption Scheme

When mounting a TrueCrypt volume (assume there are no cached passwords/keyfiles) or when performing pre-boot authentication, the following steps are performed:

1. The first 512 bytes of the volume (i.e., the standard volume header) are read into RAM, out of which the first 64 bytes are the salt (see *TrueCrypt Volume Format Specification*). For system encryption (see the chapter *System Encryption*), the last 512 bytes of the first logical drive track are read into RAM (the TrueCrypt Boot Loader is stored in the first track of the system drive and/or on the TrueCrypt Rescue Disk).

2. Bytes 65536– 66047 of the volume are read into RAM (see the section *TrueCrypt Volume Format Specification*). For system encryption, bytes 65536– 66047 of the first partition located behind the boot partition are read (see the section *Hidden Operating System*). If there is a hidden volume within this volume (or within the partition behind the boot partition), we have read its header at this point; otherwise, we have just read random data (whether or not there is a hidden volume within it has to be determined by attempting to decrypt this data; for more information see the section *Hidden Volume*).

3. Now TrueCrypt attempts to decrypt the standard volume header read in (1). All data used and generated in the course of the process of decryption are kept in RAM (TrueCrypt never saves them to disk). The following parameters are unknown[*] and have to be determined through the process of trial and error (i.e., by testing all possible combinations of the following):

    a. PRF used by the header key derivation function (as specified in PKCS #5 v2.0; see the section *Header Key Derivation, Salt, and Iteration Count*), which can be one of the following:
    HMAC-SHA-512, HMAC-RIPEMD-160, HMAC-Whirlpool.
    A password entered by the user (to which one or more keyfiles may have been applied – see the section *Keyfiles*) and the salt read in (1) are passed to the header key derivation function, which produces a sequence of values (see the section *Header Key Derivation, Salt, and Iteration Count*) from which the header encryption key and secondary header key (XTS mode) are formed. (These keys are used to decrypt the volume header.)

    b. Encryption algorithm: AES-256, Serpent, Twofish, AES-Serpent, AES-Twofish-Serpent, etc.

    c. Mode of operation: XTS, LRW (*deprecated/legacy*), CBC (*deprecated/legacy*)

    d. Key size(s)

[*] These parameters are kept secret *not* in order to increase the complexity of an attack, but primarily to make TrueCrypt volumes unidentifiable (indistinguishable from random data), which would be difficult to achieve if these parameters were stored unencrypted within the volume header. Also note that if a non-cascaded encryption algorithm is used for system encryption, the algorithm *is* known (it can be determined by analyzing the contents of the unencrypted TrueCrypt Boot Loader stored in the first logical drive track or on the TrueCrypt Rescue Disk).

4. Decryption is considered successful if the first 4 bytes of the decrypted data contain the ASCII string "TRUE", and if the CRC-32 checksum of the last 256 bytes of the decrypted data (volume header) matches the value located at byte #8 of the decrypted data (this value is unknown to an adversary because it is encrypted – see the section *TrueCrypt Volume Format Specification*). If these conditions are not met, the process continues from (3) again, but this time, instead of the data read in (1), the data read in (2) are used (i.e., possible hidden volume header). If the conditions are not met again, mounting is terminated (wrong password, corrupted volume, or not a TrueCrypt volume).

5. Now we know (or assume with very high probability) that we have the correct password, the correct encryption algorithm, mode, key size, and the correct header key derivation algorithm. If we successfully decrypted the data read in (2), we also know that we are mounting a hidden volume and its size is retrieved from data read in (2) decrypted in (3).

6. The encryption routine is reinitialized with the primary master key[*] and the secondary master key (XTS mode – see the section *Modes of Operation*), which are retrieved from the decrypted volume header (see the section *TrueCrypt Volume Format Specification*). These keys can be used to decrypt any sector of the volume, except the volume header area (or the key data area, for system encryption), which has been encrypted using the header keys. The volume is mounted.

See also section *Modes of Operation* and section *Header Key Derivation, Salt, and Iteration Count*.

[*] The master keys were generated during the volume creation and cannot be changed later. Volume password change is accomplished by re-encrypting the volume header using a new header key (derived from a new password).

Modes of Operation

The mode of operation used by TrueCrypt for encrypted partitions, drives, and virtual volumes is XTS. XTS mode is in fact XEX mode [12], which was designed by Phillip Rogaway in 2003, with a minor modification (XEX mode uses a single key for two different purposes, whereas XTS mode uses two independent keys). XTS mode was approved as the IEEE 1619 standard for cryptographic protection of data on block-oriented storage devices in December 2007.

Description of XTS mode:

$C_i = E_{K1}(P_i \wedge (E_{K2}(n) \otimes \alpha^i)) \wedge (E_{K2}(n) \otimes \alpha^i)$

Where:
$\otimes$ denotes multiplication of two polynomials over the binary field GF(2) modulo $x^{128}+x^7+x^2+x+1$
$K1$ is the encryption key (256-bit for each supported cipher; i.e., AES, Serpent, and Twofish)
$K2$ is the secondary key (256-bit for each supported cipher; i.e., AES, Serpent, and Twofish)
i is the cipher block index within a data unit; for the first cipher block within a data unit, $i = 0$
n is the data unit index within the scope of $K1$; for the first data unit, $n = 0$
α is a primitive element of Galois Field (2^{128}) that corresponds to polynomial x (i.e., 2)

The size of each data unit is always 512 bytes (regardless of the sector size).
For further information pertaining to XTS mode, see e.g. [12].

Header Key Derivation, Salt, and Iteration Count

Header key is used to encrypt and decrypt the encrypted area of the TrueCrypt volume header, which contains the master key and other data (see the sections *Encryption Scheme* and *TrueCrypt Volume Format Specification*). The method that TrueCrypt uses to generate the header key and the secondary header key (XTS mode) is PBKDF2, specified in PKCS #5 v2.0; see [7] (the document specifying PBKDF2 is also available courtesy of RSA Laboratories at: http://www.truecrypt.org/docs/pkcs5v2-0.pdf).

512-bit salt is used, which means there are 2^{512} keys for each password. This decreases vulnerability to 'off-line' dictionary attacks (pre-computing all the keys for a dictionary of passwords is very difficult when a salt is used) [7]. The salt consists of random values generated by the TrueCrypt random number generator during the volume creation process. The header key derivation function is based on HMAC-SHA-512, HMAC-RIPEMD-160, or HMAC-Whirlpool (see [8, 9, 20, 22]) – the user selects which. The length of the derived key does not depend on the size of the output of the underlying hash function. For example, a header key for the AES-256 cipher is always 256 bits long even if HMAC-RIPEMD-160 is used (in XTS mode, an additional 256-bit secondary header key is used; hence, two 256-bit keys are used for AES-256 in total). For more information, refer to [7]. 1000 iterations (or 2000 iterations when HMAC-RIPEMD-160 is used as the underlying hash function) of the key derivation function have to be performed to derive a header key, which increases the time necessary to perform an exhaustive search for passwords (i.e., brute force attack) [7].

Header keys used by ciphers in a cascade are mutually independent, even though they are derived from a single password (to which keyfiles may have been applied). For example, for the AES-Twofish-Serpent cascade, the header key derivation function is instructed to derive a 768-bit encryption key from a given password (and, for XTS mode, in addition, a 768-bit *secondary* header key from the given password). The generated 768-bit header key is then split into three 256-bit keys (for XTS mode, the *secondary* header key is split into three 256-bit keys as well, so the cascade actually uses six 256-bit keys in total), out of which the first key is used by Serpent, the second key is used by Twofish, and the third by AES (in addition, for XTS mode, the first secondary key is used by Serpent, the second secondary key is used by Twofish, and the third secondary key by AES). Hence, even when an adversary has one of the keys, he cannot use it to derive the other keys, as there is no feasible method to determine the password from which the key was derived (except for brute force attack mounted on a weak password).

Random Number Generator

The TrueCrypt random number generator (RNG) is used to generate the master encryption key, the secondary key (XTS mode), salt, and keyfiles. It creates a pool of random values in RAM (memory). The pool, which is 640 bytes long, is filled with data from the following sources:

- Mouse movements
- Keystrokes
- *Mac OS X and Linux*: Values generated by the built-in RNG (both */dev/random* and */dev/urandom*)
- *MS Windows only*: MS Windows CryptoAPI (collected regularly at 500-ms interval)
- *MS Windows only*: Network interface statistics (NETAPI32)
- *MS Windows only*: Various Win32 handles, time variables, and counters (collected regularly at 500-ms interval)

Before a value obtained from any of the above-mentioned sources is written to the pool, it is divided into individual bytes (e.g., a 32-bit number is divided into four bytes). These bytes are then individually written to the pool with the modulo 2^8 addition operation (not by replacing the old values in the pool) at the position of the pool cursor. After a byte is written, the pool cursor position is advanced by one byte. When the cursor reaches the end of the pool, its position is set to the beginning of the pool. After every 16th byte written to the pool, the pool mixing function is applied to the entire pool (see below).

Pool Mixing Function

The purpose of this function is to perform diffusion [2]. Diffusion spreads the influence of individual "raw" input bits over as much of the pool state as possible, which also hides statistical relationships. After every 16th byte written to the pool, this function is applied to the entire pool.

Description of the pool mixing function:

1. Let R be the randomness pool
2. Let H be the hash function selected by the user (SHA-512, RIPEMD-160, or Whirlpool)
3. l = byte size of the output of the hash function H (i.e., if H is RIPEMD-160, then $l = 20$; if H is SHA-512, $l = 64$)
4. z = byte size of the randomness pool R (640 bytes)
5. $q = z / l - 1$ (e.g., if H is Whirlpool, then $q = 4$)
6. R is divided into l-byte blocks $B_0 ... B_q$.
   For $0 \leq i \leq q$ (i.e., for each block B) the following steps are performed:
   a. $M = H(B_0 \| B_1 \| ... \| B_q)$ [i.e., the randomness pool is hashed using the hash function H, which produces a hash M]
   b. $B_i = B_i \wedge M$
7. $R = B_0 \| B_1 \| ... \| B_q$

   For example, if $q = 1$, the randomness pool would be mixed as follows:
   1. $(B_0 \| B_1) = R$
   2. $B_0 = B_0 \wedge H(B_0 \| B_1)$
   3. $B_1 = B_1 \wedge H(B_0 \| B_1)$
   4. $R = B_0 \| B_1$

The design and implementation of the random number generator are based on the following works:

- *Software Generation of Practically Strong Random Numbers* by Peter Gutmann [10]
- *Cryptographic Random Numbers* by Carl Ellison [11]

Keyfiles

TrueCrypt keyfile is a file whose content is combined with a password. The user can use any kind of file as a TrueCrypt keyfile. The user can also generate a keyfile using the built-in keyfile generator, which utilizes the TrueCrypt RNG to generate a file with random content (for more information, see the section *Random Number Generator*).

The maximum size of a keyfile is not limited; however, only its first 1,048,576 bytes (1 MB) are processed (all remaining bytes are ignored due to performance issues connected with processing extremely large files). The user can supply one or more keyfiles (the number of keyfiles is not limited).

Keyfiles can be stored on PKCS-11-compliant [23] security tokens and smart cards protected by multiple PIN codes (which can be entered either using a hardware PIN pad or via the TrueCrypt GUI).

Keyfiles are processed and applied to a password using the following method:

1. Let *P* be a TrueCrypt volume password supplied by user (may be empty)
2. Let *KP* be the keyfile pool
3. Let *kpl* be the size of the keyfile pool *KP*, in bytes (64, i.e., 512 bits);
   kpl must be a multiple of the output size of a hash function *H*
4. Let *pl* be the length of the password *P*, in bytes (in the current version: $0 \leq pl \leq 64$)
5. if $kpl > pl$, append $(kpl - pl)$ zero bytes to the password *P* (thus $pl = kpl$)
6. Fill the keyfile pool *KP* with *kpl* zero bytes.
7. For each keyfile perform the following steps:
   a. Set the position of the keyfile pool cursor to the beginning of the pool
   b. Initialize the hash function *H*
   c. Load all bytes of the keyfile one by one, and for each loaded byte perform the following steps:
      i. Hash the loaded byte using the hash function *H* without initializing the hash, to obtain an intermediate hash (state) *M.* Do not finalize the hash (the state is retained for next round).
      ii. Divide the state *M* into individual bytes.
      For example, if the hash output size is 4 bytes, $(T_0 \parallel T_1 \parallel T_2 \parallel T_3) = M$
      iii. Write these bytes (obtained in step 7.c.ii) individually to the keyfile pool with the modulo 2^8 addition operation (not by replacing the old values in the pool) at the position of the pool cursor. After a byte is written, the pool cursor position is advanced by one byte. When the cursor reaches the end of the pool, its position is set to the beginning of the pool.
8. Apply the content of the keyfile pool to the password *P* using the following method:
   a. Divide the password *P* into individual bytes $B_0 ... B_{pl-1}$.
   Note that if the password was shorter than the keyfile pool, then the password was padded with zero bytes to the length of the pool in Step 5 (hence, at this point the length of the password is always greater than or equal to the length of the keyfile pool).
   b. Divide the keyfile pool *KP* into individual bytes $G_0 ... G_{kpl-1}$
   c. For $0 \leq i < kpl$ perform: $B_i = B_i \oplus G_i$
   d. $P = B_0 \parallel B_1 \parallel ... \parallel B_{pl-2} \parallel B_{pl-1}$

9. The password *P* (after the keyfile pool content has been applied to it) is now passed to the header key derivation function PBKDF2 (PKCS #5 v2), which processes it (along with salt and other data) using a cryptographically secure hash algorithm selected by the user (e.g., RIPEMD-160 or Whirlpool). See the section *Header Key Derivation, Salt, and Iteration Count* for more information.

The role of the hash function *H* is merely to perform diffusion [2]. CRC-32 is used as the hash function *H*. Note that the output of CRC-32 is subsequently processed using a cryptographically secure hash algorithm: The keyfile pool content (in addition to being hashed using CRC-32) is applied to the password, which is then passed to the header key derivation function PBKDF2 (PKCS #5 v2), which processes it (along with salt and other data) using a cryptographically secure hash algorithm selected by the user (e.g., RIPEMD-160 or Whirlpool). The resultant values are used to form the header key and the secondary header key (XTS mode).

TrueCrypt Volume Format Specification

Offset (bytes)	Size (bytes)	Encryption Status[*]	Description
0	64	Unencrypted[‡]	Salt
64	4	Encrypted	ASCII string “TRUE”
68	2	Encrypted	Volume header format version
70	2	Encrypted	Minimum program version required to open the volume
72	4	Encrypted	CRC-32 checksum of the (decrypted) bytes 256–511
76	16	Encrypted	Reserved (set to zero)
92	8	Encrypted	Size of hidden volume (for normal volumes, set to zero)
100	8	Encrypted	Size of volume
108	8	Encrypted	Byte offset of the start of the master key scope
116	8	Encrypted	Size of the encrypted area within the master key scope
124	4	Encrypted	Flag bits (bit 0 set: system encryption; bits 1–31 are reserved)
128	124	Encrypted	Reserved (set to zero)
252	4	Encrypted	CRC-32 checksum of the (decrypted) bytes 64–251
256	*Var.*	Encrypted	Concatenated primary and secondary master keys[§]
512	65024	Encrypted	Reserved (for system encryption, this item is omitted[††])
65536	65536	Encrypted / Unencrypted[‡]	Area for hidden volume header (if there is no hidden volume within the volume, this area contains random data[**]). For system encryption, this item is omitted.[††] See bytes 0–65535.
131072	*Var.*	Encrypted	Data area (master key scope). For system encryption, offset may be different (depending on offset of system partition).
S-131072[†]	65536	Encrypted / Unencrypted[‡]	Backup header (encrypted with a different header key derived using a different salt). For system encryption, this item is omitted.[††] See bytes 0–65535.
S-65536	65536	Encrypted / Unencrypted[‡]	Backup header for hidden volume (encrypted with a different header key derived using a different salt). If there is no hidden within the volume, this area contains random data.[**] For system encryption, this item is omitted.[††] See bytes 0–65535.

[*] The encrypted areas of the volume header are encrypted with the header key (and the secondary header key in XTS mode). For more information, see the section *Encryption Scheme* and the section *Header Key Derivation, Salt, and Iteration Count*.
[†] *S* denotes the size of the volume (in bytes).
[‡] Note that the salt does not need to be encrypted, as it does not have to be kept secret [7] (salt is a sequence of random values).
[§] Multiple concatenated master keys are stored here when the volume is encrypted using a cascade of ciphers (secondary master keys are used for XTS mode).
[**] See below in this section for information on the method used to fill free volume space with random data when the volume is created.
[††] Here, the meaning of "system encryption" does not include a hidden volume containing a hidden operating system.

The format of file-hosted volumes is identical to the format of partition/device-hosted volumes (however, the "volume header", or key data, for a system partition/drive is stored in the last 512 bytes of the first logical drive track). TrueCrypt volumes have no "signature" or ID strings. Until decrypted, they appear to consist of random data. Therefore, it is impossible to identify a TrueCrypt container or partition.

Free space on each TrueCrypt volume is filled with random data when the volume is created.[*] The random data is generated as follows: Right before TrueCrypt volume formatting begins, a temporary encryption key and a temporary secondary key (XTS mode) are generated by the random number generator (see the section *Random Number Generator*). The encryption algorithm that the user selected is initialized with the temporary keys. The encryption algorithm is then used to encrypt plaintext blocks generated by the random number generator. The encryption algorithm operates in XTS mode (see the section *Modes of Operation*). The resulting ciphertext blocks are used to fill (overwrite) the free space on the volume. The temporary keys are stored in RAM and are securely erased after formatting finishes.
The fields located at the byte #0 (salt) and #256 (master keys) contain random values generated by the random number generator (see the section *Random Number Generator*) during the volume creation process.

If a TrueCrypt volume hosts a hidden volume (within its free space), the header of the hidden volume is located at the byte #65536 of the host volume (the header of the host/outer volume is located at the byte #0 of the host volume – see the section *Hidden Volume*). If there is no hidden volume within a TrueCrypt volume, the bytes 65536–131071 of the volume (i.e., the area where the header of a hidden volume can reside) contain random data (see above for information on the method used to fill free volume space with random data when the volume is created). The layout of the header of a hidden volume is the same as the one of a standard volume (bytes 0–65535).

The maximum possible TrueCrypt volume size is 2^{63} bytes (8,589,934,592 GB). However, due to security reasons (with respect to the 128-bit block size and the mode of operation), the maximum allowed volume size is 1 PB (1,048,576 GB).

Embedded Backup Headers

Each TrueCrypt volume created by TrueCrypt 6.0 or later contains an embedded backup header, located at the end of the volume (see above). The header backup is *not* a copy of the volume header because it is encrypted with a different header key derived using a different salt (see the section *Header Key Derivation, Salt, and Iteration Count*).

When the volume password and/or keyfiles are changed, or when the header is restored from the embedded (or an external) header backup, both the volume header and the backup header (embedded in the volume) are re-encrypted with different header keys (derived using newly generated salts – the salt for the volume header is different from the salt for the backup header). Each salt is generated by the TrueCrypt random number generator (see the section *Random Number Generator*).

For more information about header backups, see the subsection *Tools -> Restore Volume Header* in the chapter *Main Program Window*.

[*] Provided that the options *Quick Format* and *Dynamic* are disabled and provided that the volume does not contain a filesystem that has been encrypted in place (note that TrueCrypt does not allow the user to create a hidden volume within such a volume).

Compliance with Standards and Specifications

TrueCrypt complies with the following standards, specifications, and recommendations:

- PKCS #5 v2.0 [7]
- PKCS #11 v2.20 [23]
- FIPS 197 [3]
- FIPS 198 [22]
- FIPS 180-2 [14]
- ISO/IEC 10118-3:2004 [21]

The correctness of the implementations of the encryption algorithms can be verified using test vectors (select *Tools* > *Test Vectors*) or by examining the source code of TrueCrypt.

Source Code

TrueCrypt is open-source and free software. The complete source code of TrueCrypt (written in C, C++, and assembly) is freely available for peer review at:

http://www.truecrypt.org/

Future Development

For the list of features that are planned for a future release, please refer to:
http://www.truecrypt.org/future.php

License

The text of the license under which TrueCrypt is distributed is contained in the file *License.txt* that is included in the TrueCrypt binary and source code distribution packages, and is also available at:
http://www.truecrypt.org/legal/license

Contact

Information on how to contact us can be found at:
http://www.truecrypt.org/contact.php

Version History

6.1a

December 1, 2008

Improvements, bug fixes, and security enhancements:

- Minor improvements, bug fixes, and security enhancements. *(Windows, Mac OS X, and Linux)*

  Note: If you are using an older version of TrueCrypt, it is highly recommended that you upgrade to the latest stable version.

6.1

October 31, 2008

New features:

- Ability to encrypt a non-system partition without losing existing data on the partition. (*Windows Vista/2008*)

  Note: To encrypt a non-system partition in place, click '*Create Volume*' > '*Encrypt a non-system partition*' > '*Standard volume*' > '*Select Device*' > '*Encrypt partition in place*' and then follow the instructions in the wizard. Please note that this is not supported on Windows XP/2000/2003 as these versions of Windows do not natively support shrinking of a filesystem (the filesystem needs to be shrunk to make space for the volume header and backup header).

- Support for security tokens and smart cards (for more information, see section *Security Tokens and Smart Cards* in chapter *Keyfiles*).

- The TrueCrypt boot loader can be prevented from displaying any texts (by selecting *Settings* > *System Encryption* and enabling the option '*Do not show any texts in the pre-boot authentication screen'*).

- The TrueCrypt boot loader can now display a custom message (select *Settings* > *System Encryption* and enter the message in the corresponding field) either without any other texts or along with the standard TrueCrypt boot loader texts.

- Pre-boot authentication passwords can now be cached in the driver memory, which allows them to be used for mounting of non-system TrueCrypt volumes (select *Settings* > *System Encryption* and enable the option '*Cache pre-boot authentication password*').

- *Linux and Mac OS X versions*: The ability to mount a Windows system partition encrypted by TrueCrypt and to mount a partition located on a Windows system drive that is fully encrypted by a Windows version of TrueCrypt.

Improvements:

- Protection against memory corruption caused by bugs in certain versions of some BIOSes, which prevented the TrueCrypt boot loader from working properly. (*Windows Vista/XP/2008/2003*)

- During the process of creation of a hidden operating system, TrueCrypt now securely erases the entire content of the partition where the original system resides after the hidden system has been created. The user is then prompted to install a new system on the partition and encrypt it using TrueCrypt (thus the decoy system is created).

Note: Although we are not aware of any security issues (connected with decoy systems) affecting the previous versions of TrueCrypt, we have implemented this change to prevent any such undiscovered security issues (if there are any). Otherwise, in the future, a vulnerability might be discovered that could allow an attacker to find out that the TrueCrypt wizard was used in the hidden-system-creation mode (which might indicate the existence of a hidden operating system on the computer) e.g. by analyzing files, such as log files created by Windows, stored on the partition where the original system (of which the hidden system is a clone) resides. In addition, due to this change, it is no longer required that the paging file is disabled and hibernation prevented when creating a hidden operating system.

- Many other improvements. (*Windows, Mac OS X, and Linux*)

Bug fixes:

- Many minor bug fixes and security improvements. (*Windows, Mac OS X, and Linux*)

For a list of changes in older versions, see: http://www.truecrypt.org/docs/?s=version-history

Acknowledgements

We would like to thank the following people:

Paul Le Roux for making his E4M source code available; some parts of TrueCrypt were derived from E4M.

Brian Gladman, who wrote the excellent AES, Twofish, SHA-512, and finite field GF(2^{128}) multiplication routines.

Peter Gutmann for his paper on random numbers, and for creating his cryptlib, which was the source of parts of the random number generator source code.

Wei Dai, who wrote the Serpent and RIPEMD-160 routines, and *Dag Arne Osvik* for his paper *Speeding up Serpent*.

Mark Adler et al., who wrote the Inflate routine.

The designers of the encryption algorithms, hash algorithms, and the mode of operation:
Horst Feistel, Don Coppersmith, Walt Tuchmann, Lars Knudsen, Ross Anderson, Eli Biham, Bruce Schneier, David Wagner, John Kelsey, Niels Ferguson, Doug Whiting, Chris Hall, Joan Daemen, Vincent Rijmen, Carlisle Adams, Stafford Tavares, *Phillip Rogaway, Hans Dobbertin, Antoon Bosselaers*, *Bart Preneel*, *Paulo S. L. M. Barreto.*

All the others who have made this project possible, all who have morally supported us, and all who sent us bug reports or suggestions for improvements.

Thank you very much.

References

[1] U.S. Committee on National Security Systems (CNSS), *National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information*, CNSS Policy No. 15, Fact Sheet No. 1, June 2003, available at http://www.cnss.gov/Assets/pdf/cnssp_15_fs.pdf and also at http://csrc.nist.gov/cryptval/CNSS15FS.pdf.

[2] C. E. Shannon, *Communication Theory of Secrecy Systems*, Bell System Technical Journal, v. 28, n. 4, 1949

[3] NIST, *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197, November 26, 2001, available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[4] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, E. Roback, NIST, *Report on the Development of the Advanced Encryption Standard (AES)*, October 2, 2000, available at http://nvl.nist.gov/pub/nistpubs/jres/106/3/j63nec.pdf.

[5] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, T. Kohno, M. Stay, *The Twofish Team's Final Comments on AES Selection,* May 15, 2000, available at http://csrc.nist.gov/CryptoToolkit/aes/round2/comments/20000515-bschneier.pdf.

[6] M. Bellare, *New Proofs for NMAC and HMAC: Security without Collision-Resistance*, Cryptology ePrint Archive: Report 2006/043, February 6, 2006, available at http://eprint.iacr.org/2006/043

[7] RSA Laboratories, *PKCS #5 v2.0: Password-Based Cryptography Standard*, RSA Data Security, Inc. Public-Key Cryptography Standards (PKCS), March 25, 1999, available at ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf
and also courtesy of RSA Laboratories at: http://www.truecrypt.org/docs/pkcs5v2-0.pdf

[8] H. Krawczyk, M. Bellare, R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, RFC 2104, February 1997, available at http://www.ietf.org/rfc/rfc2104.txt.

[9] M. Nystrom, RSA Security, *Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512*, RFC 4231, December 2005, available at http://www.ietf.org/rfc/rfc4231.txt.

[10] Peter Gutmann, *Software Generation of Practically Strong Random Numbers*, presented at the 1998 Usenix Security Symposium, available at http://www.cs.auckland.ac.nz/~pgut001/pubs/usenix98.pdf.

[11] Carl Ellison, *Cryptographic Random Numbers*, originally an appendix to the P1363 standard, available at http://world.std.com/~cme/P1363/ranno.html.

[12] P. Rogaway, *Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC*, Asiacrypt 2004. LNCS vol. 3329. Springer, 2004. Also available at: http://www.cs.ucdavis.edu/~rogaway/papers/offsets.pdf.

[13] J. Kelsey, *Twofish Technical Report #7: Key Separation in Twofish,* AES Round 2 public comment, April 7, 2000

[14] NIST, *Secure Hash Standard*, FIPS 180-2, August 1, 2002, available at http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.

[15] U. Maurer, J. Massey, *Cascade Ciphers: The Importance of Being First*, Journal of Cryptology, v. 6, n. 1, 1993

[16] Bruce Schneier, *Applied Cryptography*, Second Edition, John Wiley & Sons, 1996

[17] Peter Gutmann, *Secure Deletion of Data from Magnetic and Solid-State Memory*, first published in the Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996, available at http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html

[18] Serpent home page: http://www.cl.cam.ac.uk/~rja14/serpent.html.

[19] M. E. Smid, *AES Issues*, AES Round 2 Comments, May 22, 2000, available at http://csrc.nist.gov/CryptoToolkit/aes/round2/comments/20000523-msmid-2.pdf.

[20] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, October 1996

[21] International Organization for Standardization (ISO), *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*, ISO/IEC 10118-3:2004, February 24, 2004

[22] NIST, *The Keyed-Hash Message Authentication Code (HMAC)*, Federal Information Processing Standards Publication 198, March 6, 2002, available at http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf.

[23] RSA Laboratories, *PKCS #11 v2.20: Cryptographic Token Interface Standard*, RSA Security, Inc. Public-Key Cryptography Standards (PKCS), June 28, 2004, available at ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf.