TRUECRYPT

FREE OPEN-SOURCE ON-THE-FLY ENCRYPTION

USER'S GUIDE

www.truecrypt.org

Version Information

TrueCrypt User's Guide, version 4.3. Released March 19, 2007.

Licensing Information

By installing and/or running and/or using TrueCrypt you agree to the license contained in the file *License.txt*, which is included in TrueCrypt binary and source code distribution packages.

Copyright Information

A TrueCrypt Foundation Release

Trademark Information

Limitations

The TrueCrypt Foundation does not warrant that the information contained in this document meets your requirements or that the information is free of errors. The information may include technical inaccuracies or typographical errors.

CONTENTS

PREFACE

Please note that although many chapters of this document (such as *Technical Details* and *Plausible Deniability*) 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 Linux versions of TrueCrypt. Linux-specific features are described in the TrueCrypt man page, which is included in the TrueCrypt binary and source code distribution packages, which are available at: http://www.truecrypt.org/downloads.php.

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. The 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.

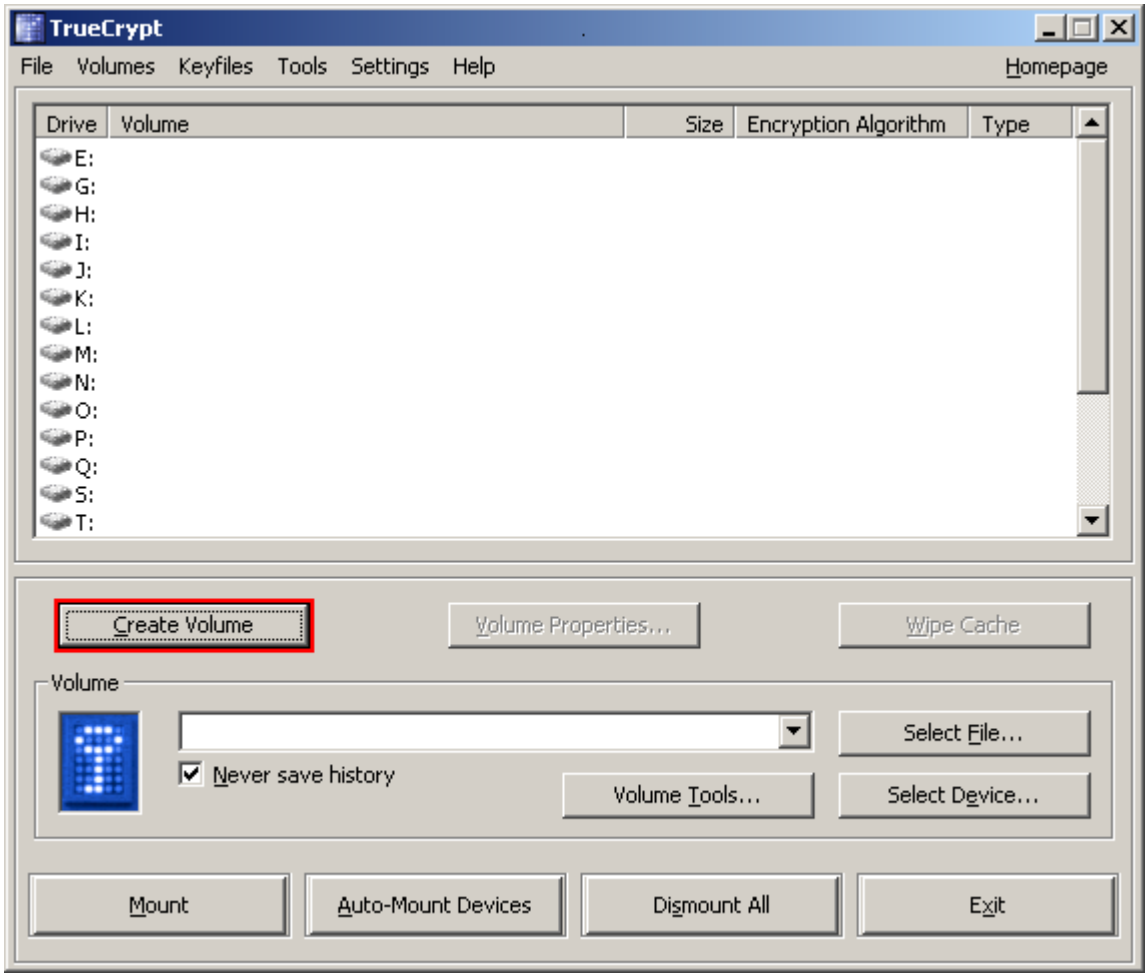
STEP 1:

If you have not done so, download, unpack, and install TrueCrypt (to do so, double-click *TrueCrypt Setup.exe* and then click **Install**).

STEP 2:

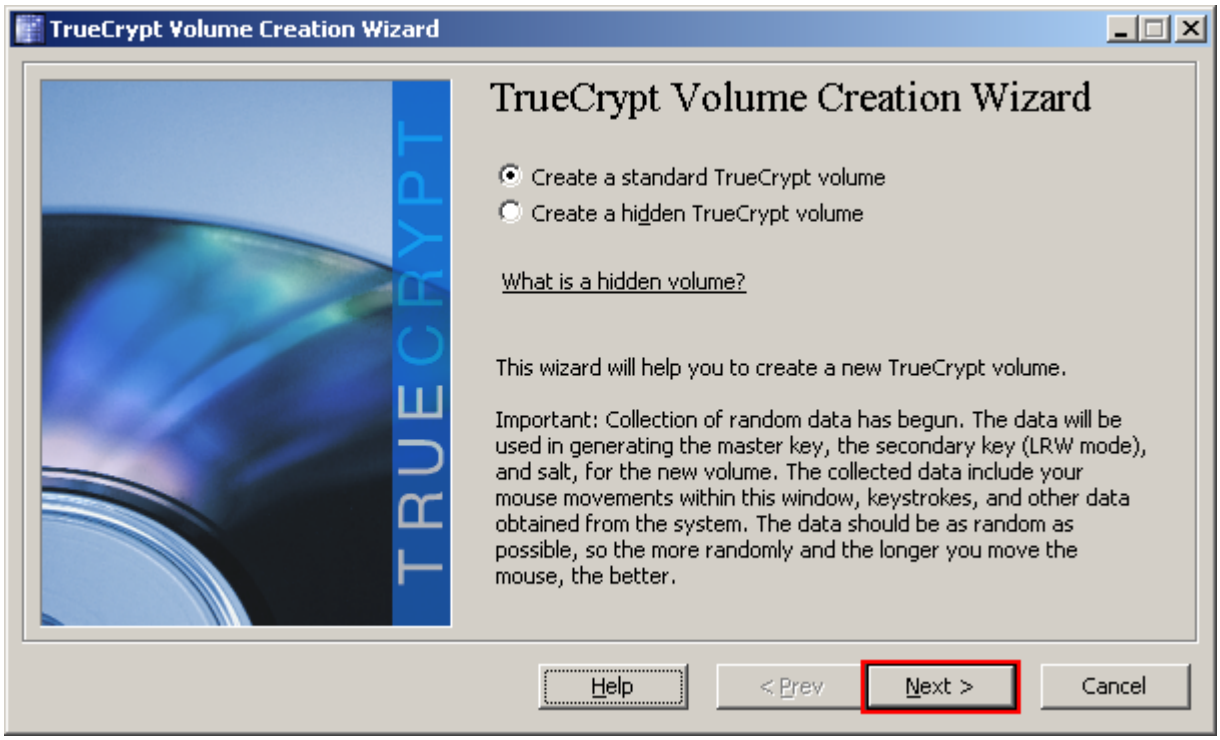
Launch TrueCrypt by double-clicking the file *TrueCrypt.exe* or by clicking the TrueCrypt shortcut in your Windows Start menu.

STEP 3:



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

STEP 4:



The TrueCrypt Volume Creation Wizard window should appear.

Read the instructions displayed in the Wizard window and click **Next**.

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

STEP 5:

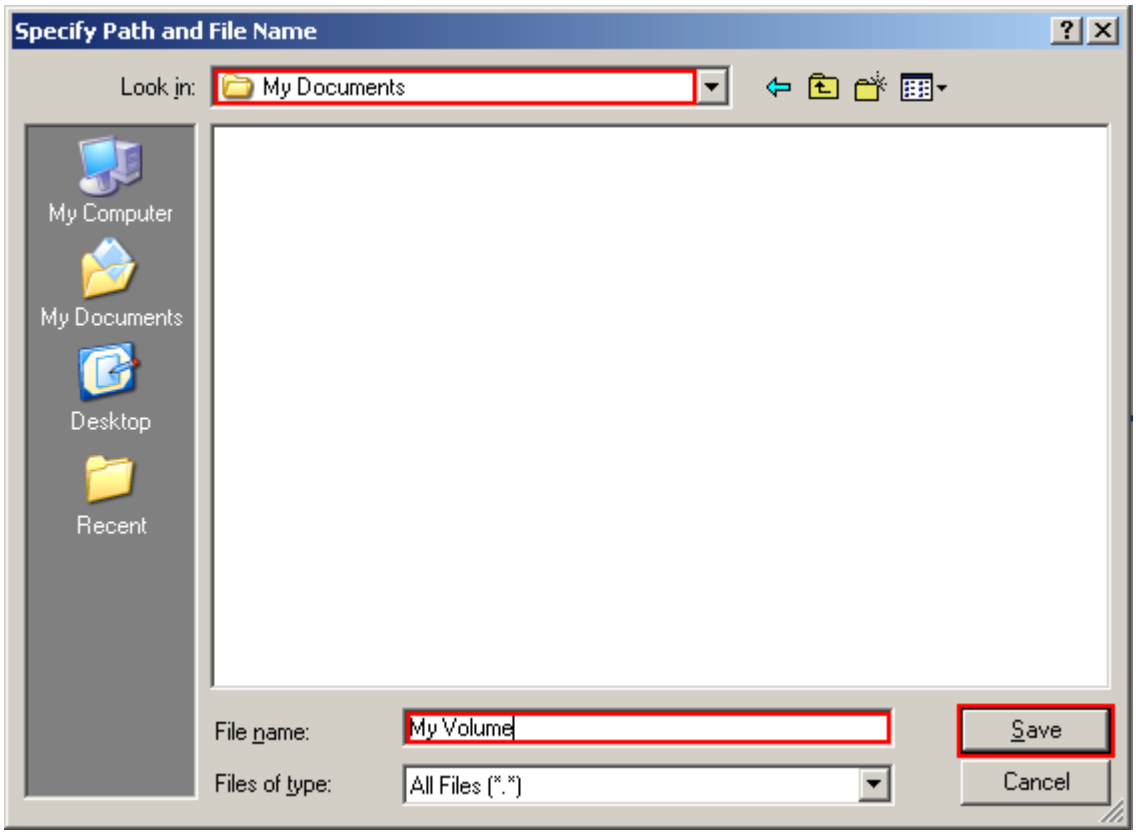
In this step you have to specify where you wish the TrueCrypt volume to be created. A TrueCrypt volume can reside either in a file, which is also called *container*, or in a partition (device). In this tutorial, we will choose the former option and create a TrueCrypt volume within a file.

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.

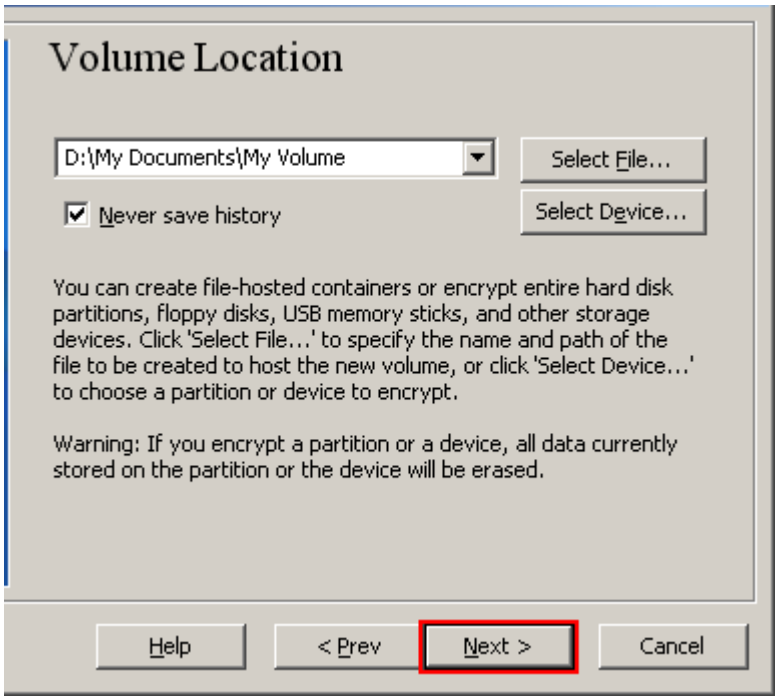
Click **Save**.

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:

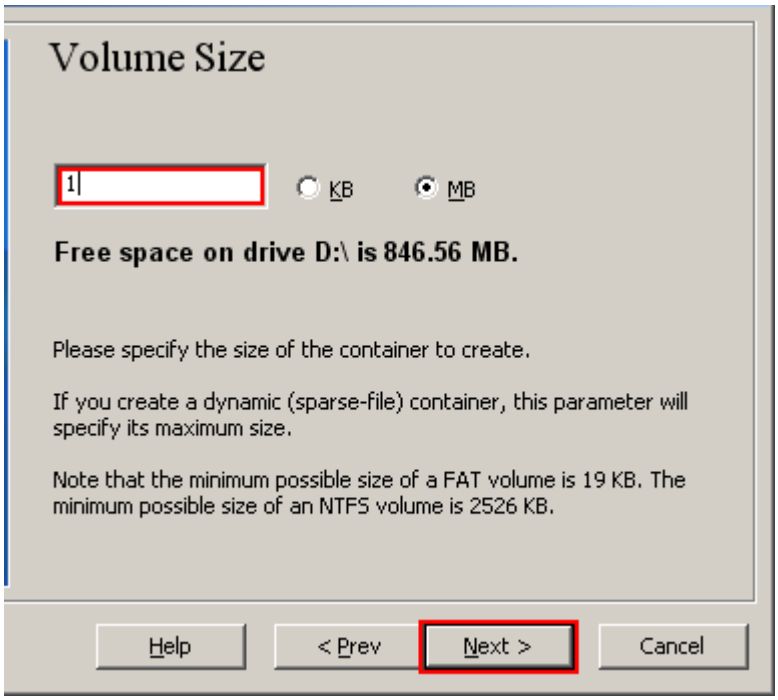


In the Volume Creation Wizard window, click **Next**.

STEP 8:

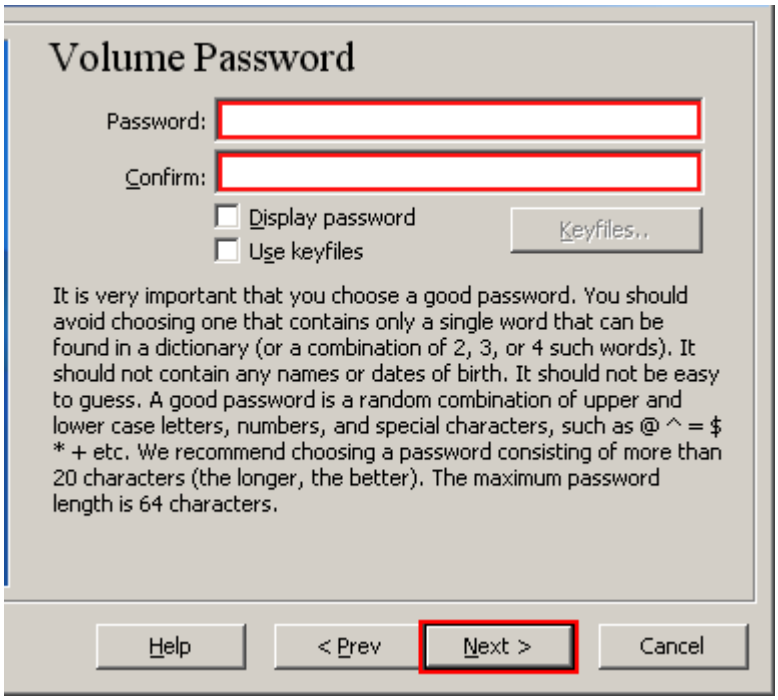
Here you can choose an encryption algorithm and a hash algorithm for the volume. If you are not sure what to select here, you can use the default settings and click **Next** (for more information, see Chapters *Encryption Algorithms* and *Hash Algorithms*).

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:



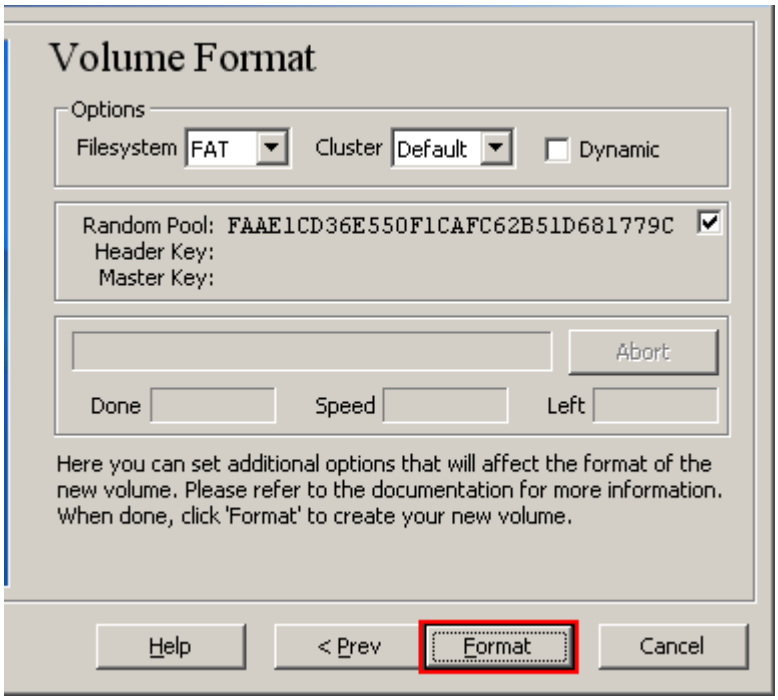
This is one of the most important steps. Here you have to choose a good volume password.

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:



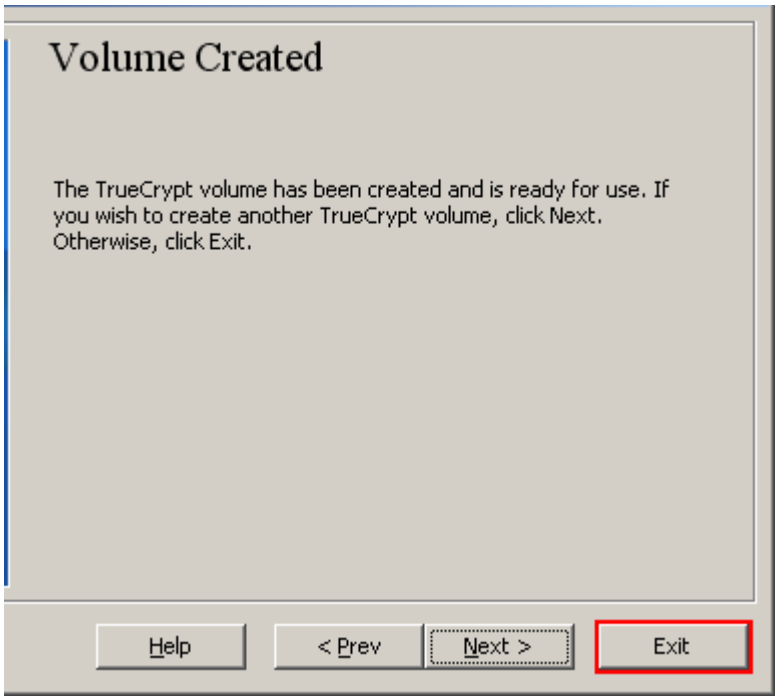
Move your mouse as randomly as possible within the Volume Creation Wizard window at least for 30 seconds. The longer you move the mouse, the better. This is important for the quality of the encryption key.

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:



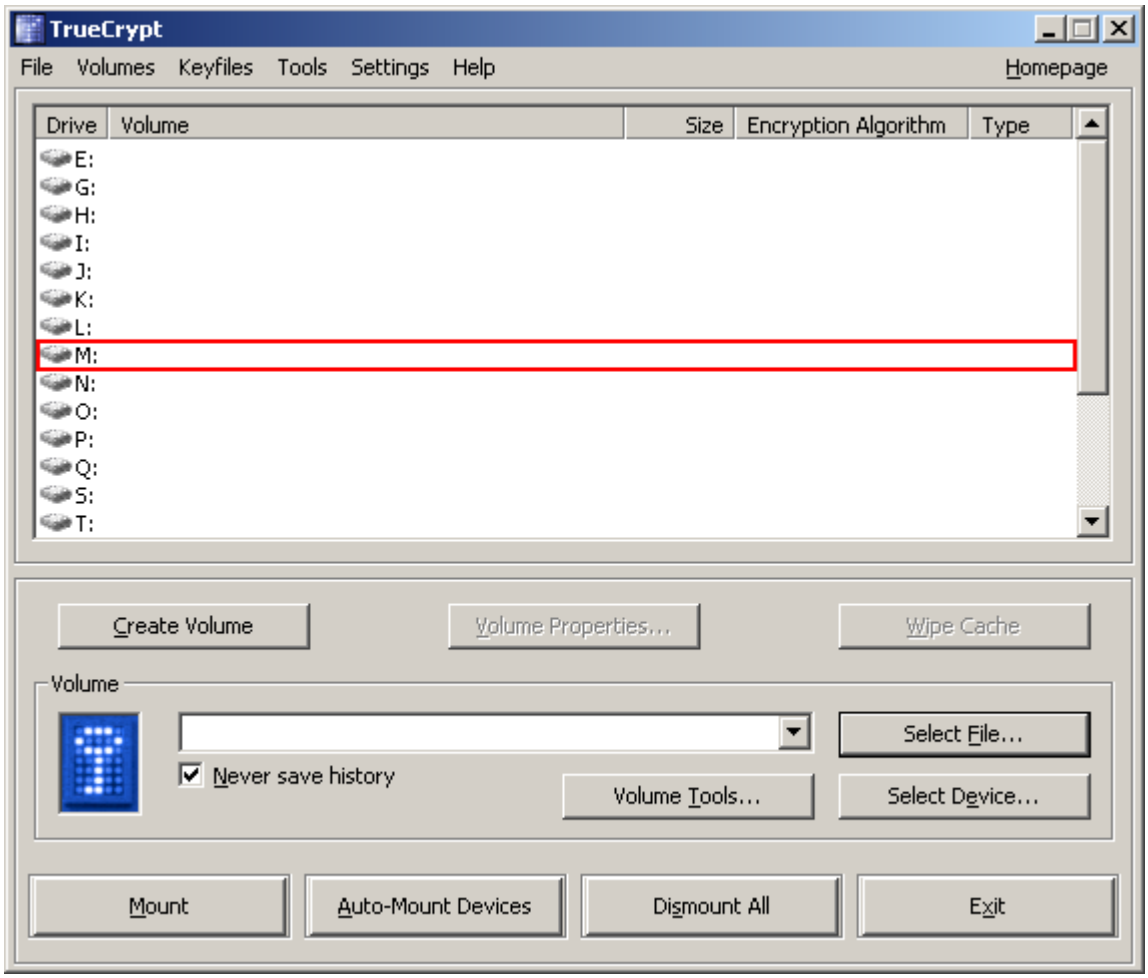
We have just successfully created a TrueCrypt volume (file container).

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. (It should still be open, but if it is not, repeat Step 2 to launch TrueCrypt and then continue from Step 13.)

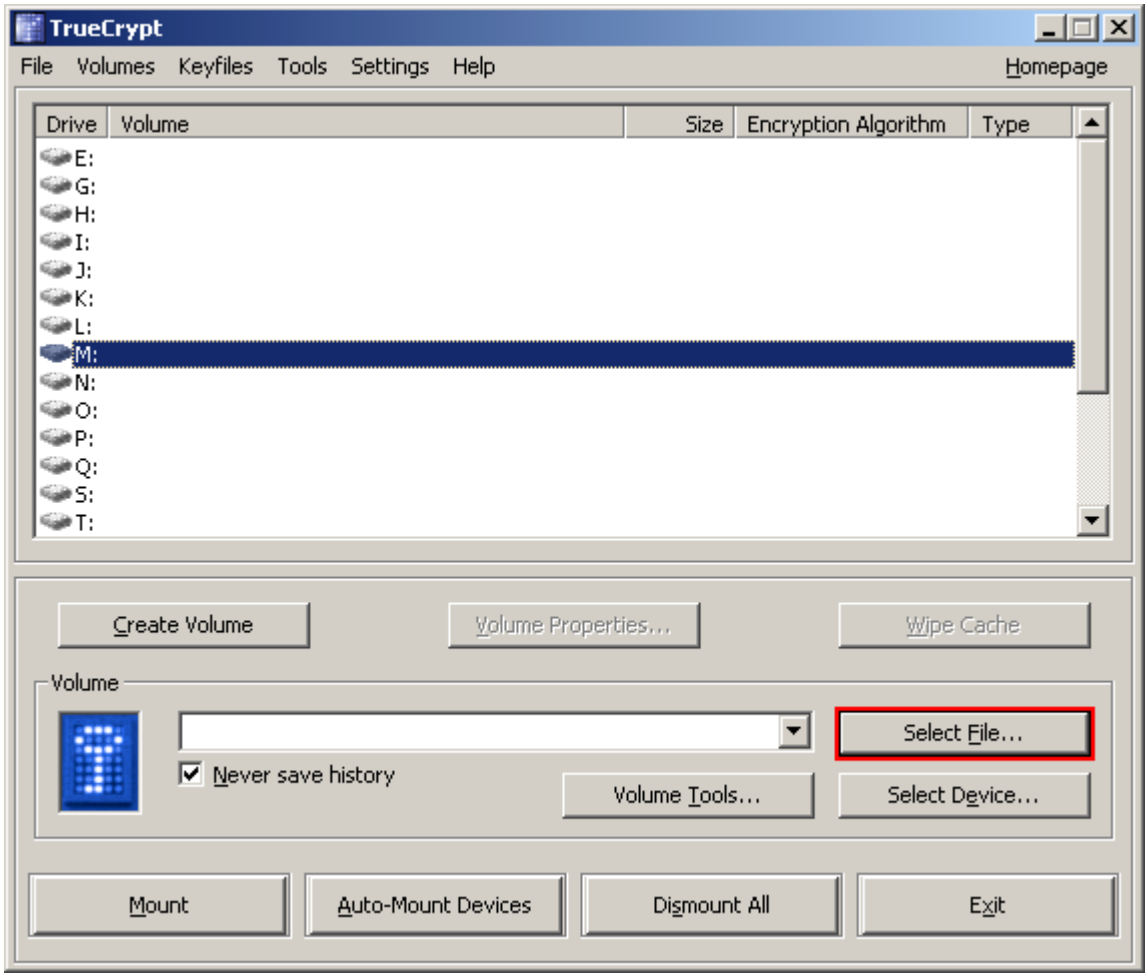
STEP 13:



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.

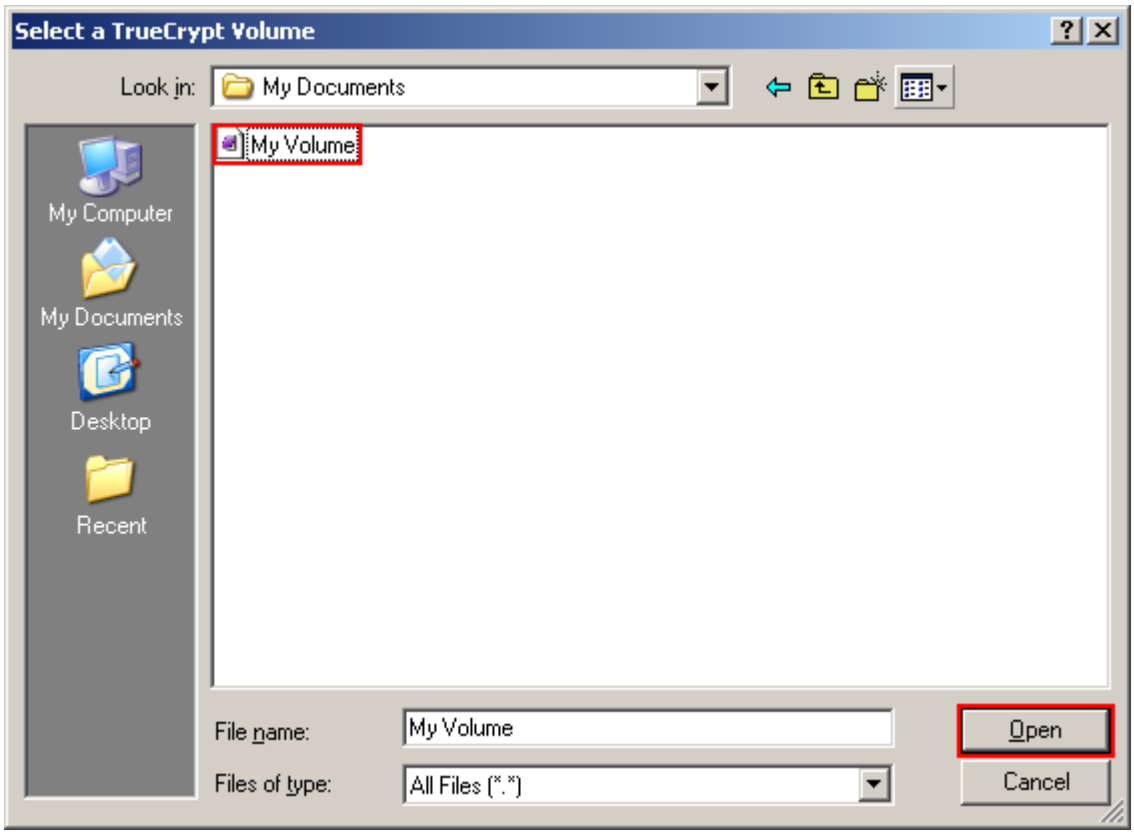
STEP 14:



Click **Select File**.

The standard file selector window should appear.

STEP 15:



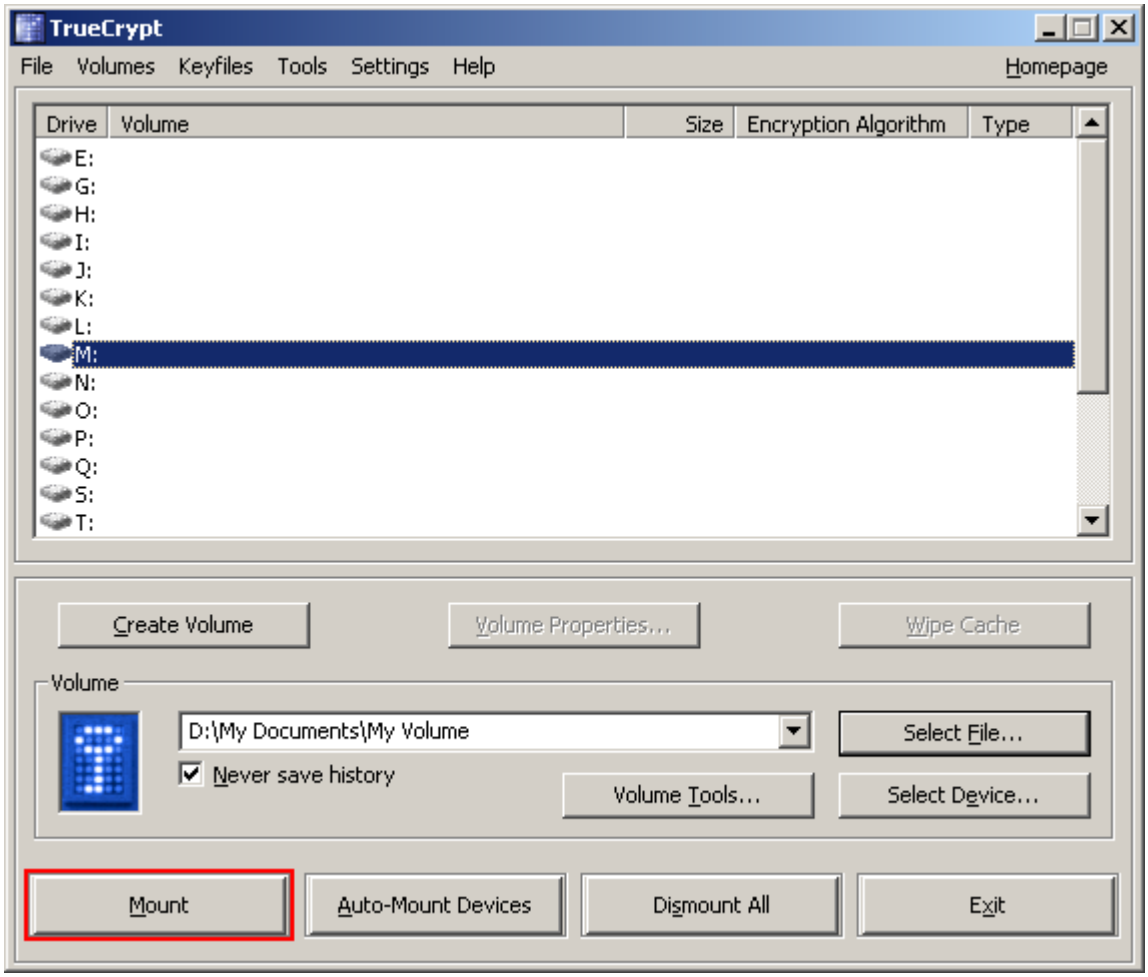
In the file selector, browse to the container file (which we created in Steps 6-11) and select it.

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.

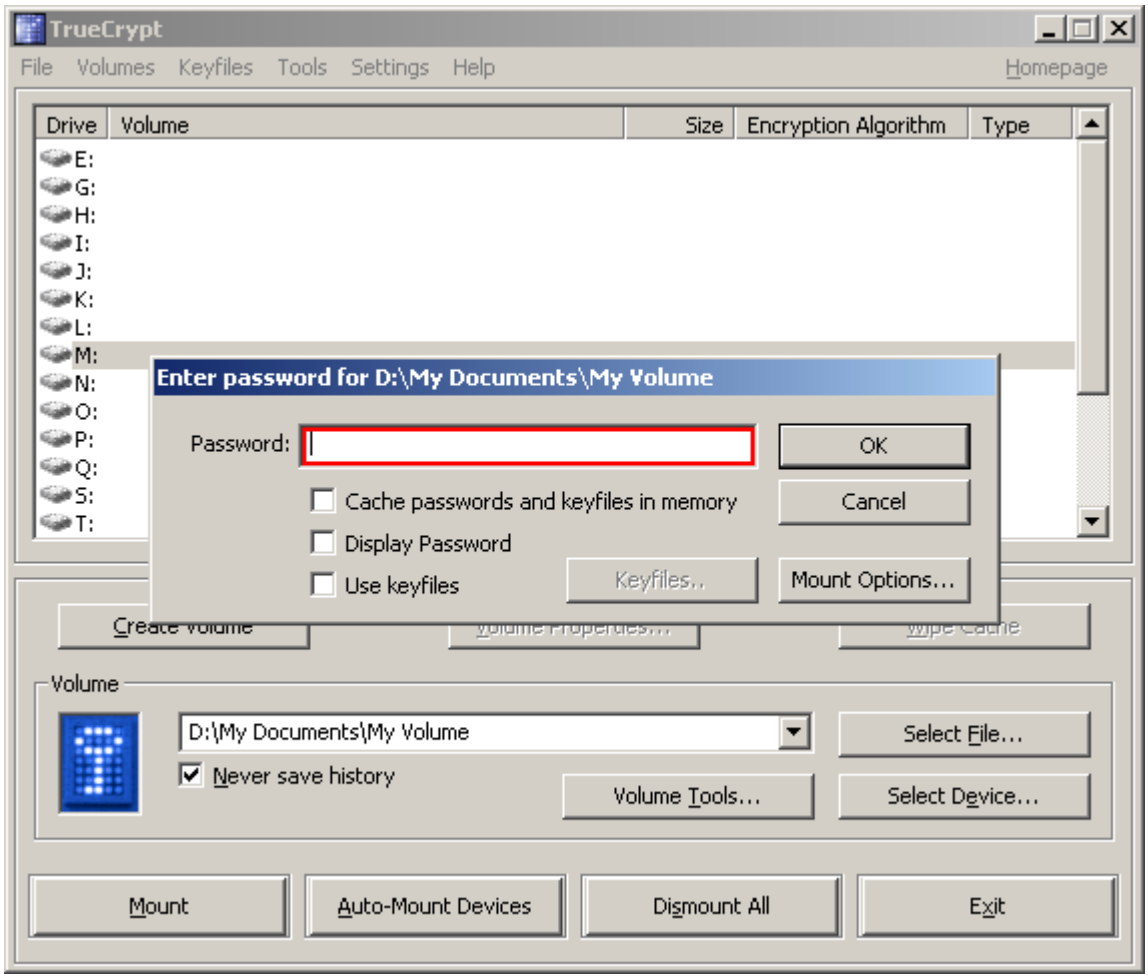
STEP 16:



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

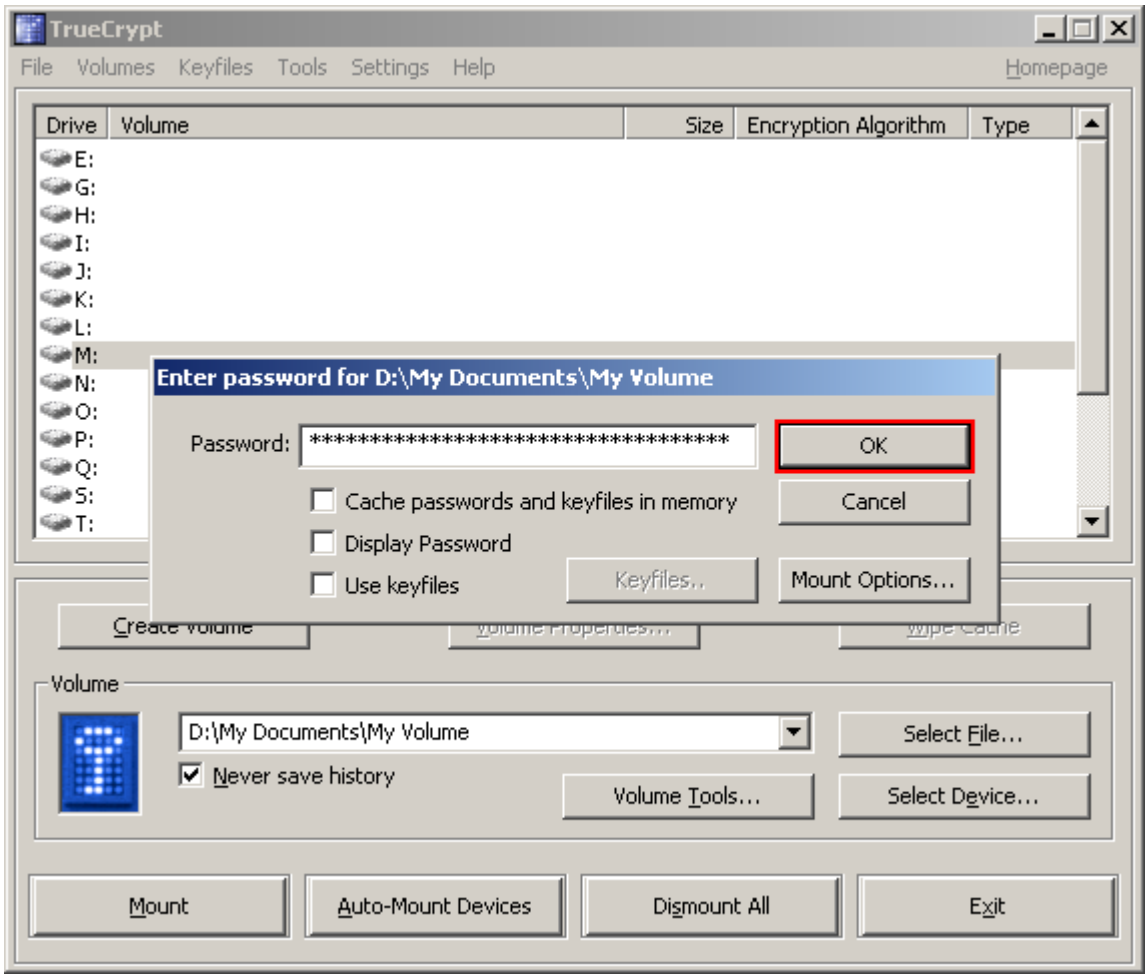
Password prompt dialog window should appear.

STEP 17:



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

STEP 18:

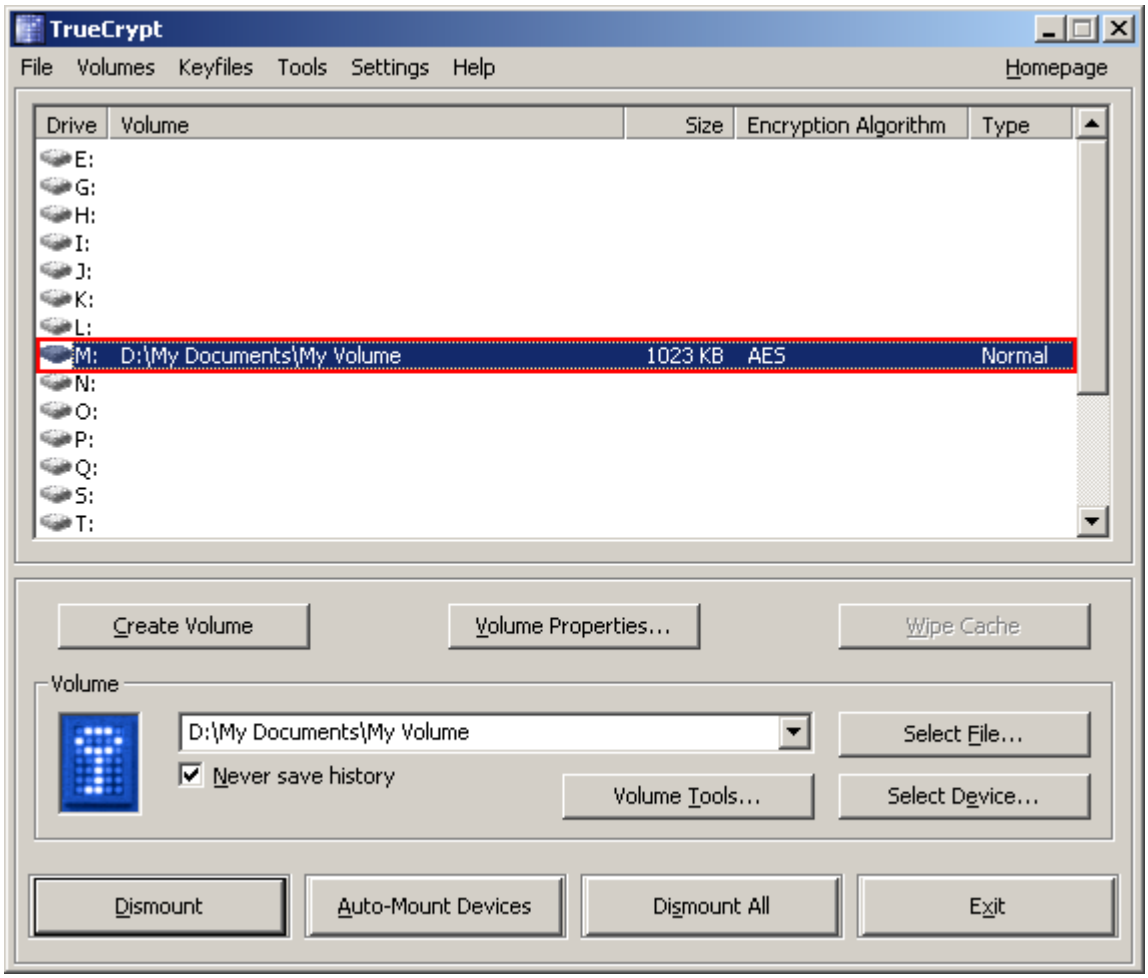


Click **OK** in the password prompt window.

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.

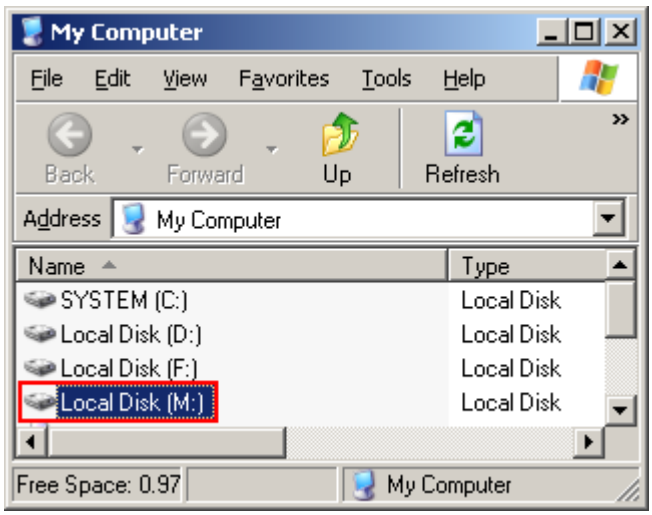
If you open a file stored on a TrueCrypt volume, for example, in media player, the file will be automatically decrypted to RAM (memory) on-the-fly while it is being read.

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).

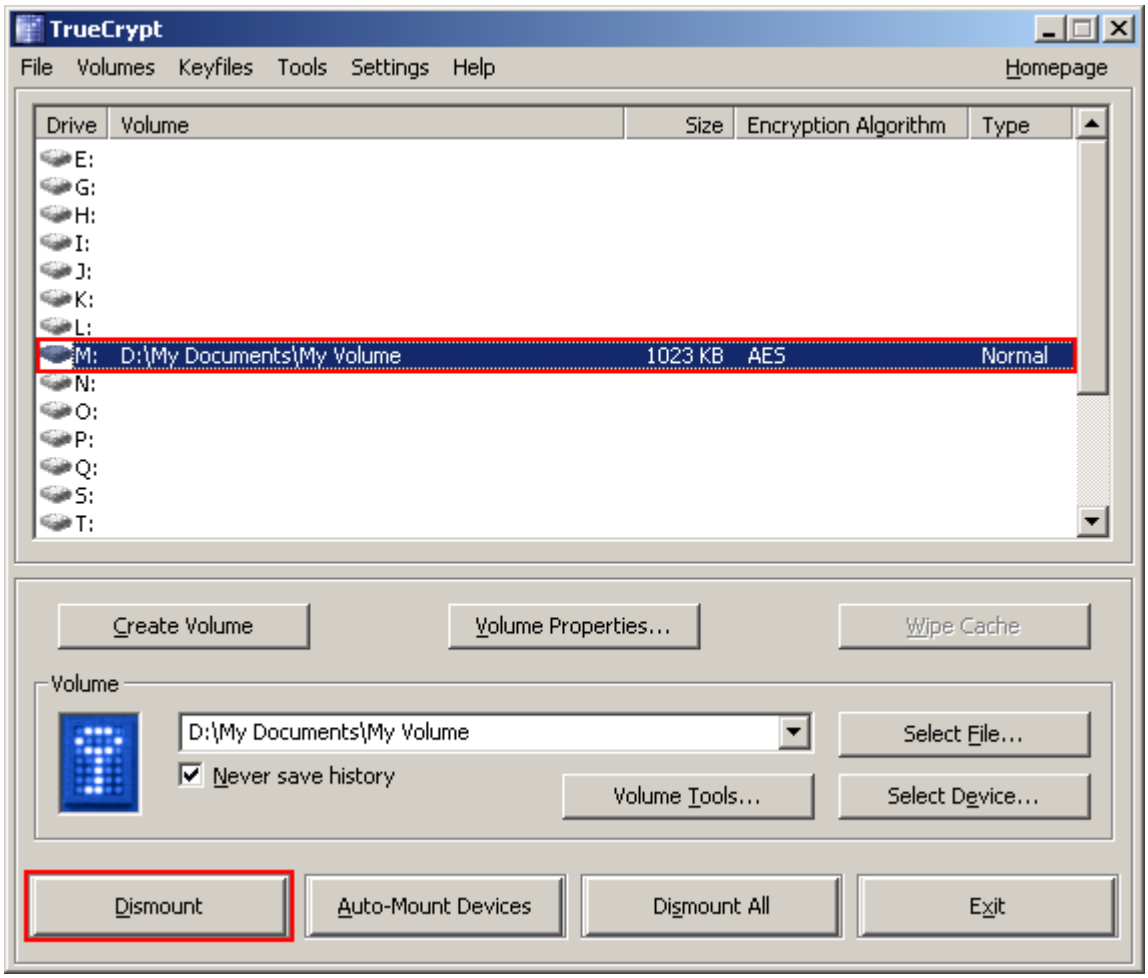


You can copy files to and from the TrueCrypt volume just as you would copy them to any normal disk (for example, by simple drag-and-drop operations). Files that are being read or copied from the encrypted TrueCrypt volume are automatically decrypted on-the-fly (in memory/RAM). Similarly, files that are 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 devices (i.e., create TrueCrypt device-hosted volumes). To do so, repeat the steps 1-18 described in the previous section of this tutorial, but, in all relevant steps, instead of clicking **Select File,** click **Select Device**.

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* bellow).

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.

TrueCrypt containers (file-hosted volumes) can have any file extension you like (for example, .raw, .iso, .bin, .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). We also recommend that you avoid file extensions used for executable files, such as .exe, .sys, and .dll. Otherwise, Windows and antivirus software may interfere with the container and adversely affect the performance of the volume.

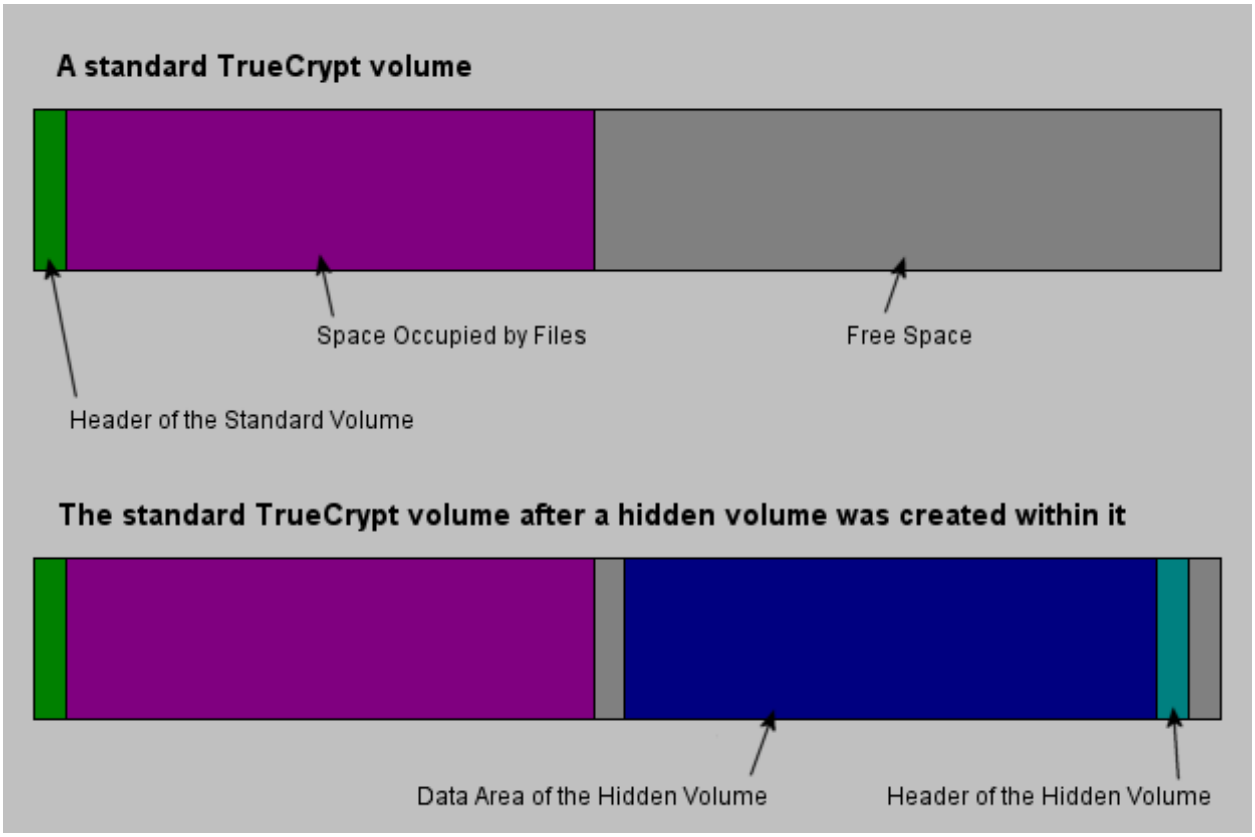
When formatting a hard disk partition as a TrueCrypt volume, the partition table (including the partition type) is *never* modified (no TrueCrypt "signature" or "ID" is written to the partition table).

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 behaviour 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.

* Provided that the options *Quick Format* and *Dynamic* are disabled. For information on the method used to fill free volume space with random data, see chapter *Technical Details*, section *TrueCrypt Volume Format Specification.*

The password for the hidden volume must be 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 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 sector of the volume where hidden volume headers are normally stored (the third sector from the end of the volume) to RAM and attempts to decrypt it using the entered password. Note that 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).

A hidden volume can be created within any type of TrueCrypt volume, i.e., within a file-hosted volume or within a partition/device (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.*

A hidden volume can only be created within a FAT TrueCrypt volume (i.e., the file system of the outer volume must either be FAT12, FAT16, or FAT32). NTFS file system stores various data throughout the entire volume (as opposed to FAT) leaving little room for the hidden volume. Therefore, the Volume Creation Wizard prevents the user from selecting NTFS as the file system for the outer volume. The hidden volume can contain any file system you like. Note that the outer volume (when file-hosted) can be stored on any file system.
Note: Should you be asked why the file system of the outer volume is FAT, you can answer that you left all settings at default (FAT is the default file system for all TrueCrypt volumes). There are also other reasons to use FAT instead of NTFS (for example, FAT is faster and tends to get less fragmented).

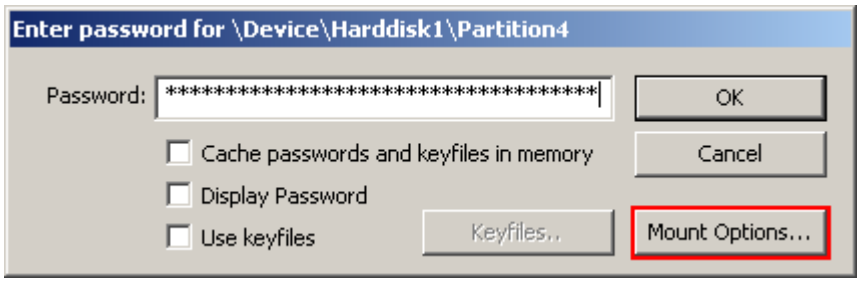
If there are any problems when creating a hidden volume, refer to the chapter *Troubleshooting* for possible solutions.

* This feature is implemented only in the Windows versions of TrueCrypt. 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.

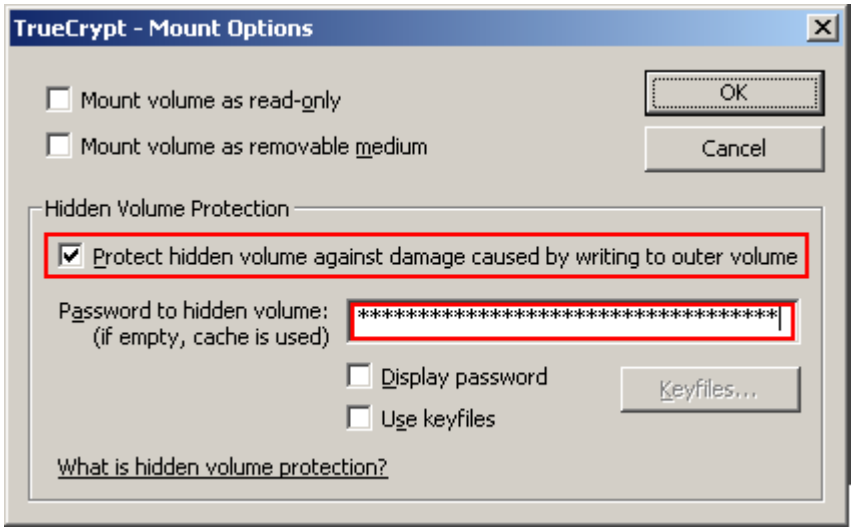
Protection of Hidden Volumes Against Damage

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 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*:



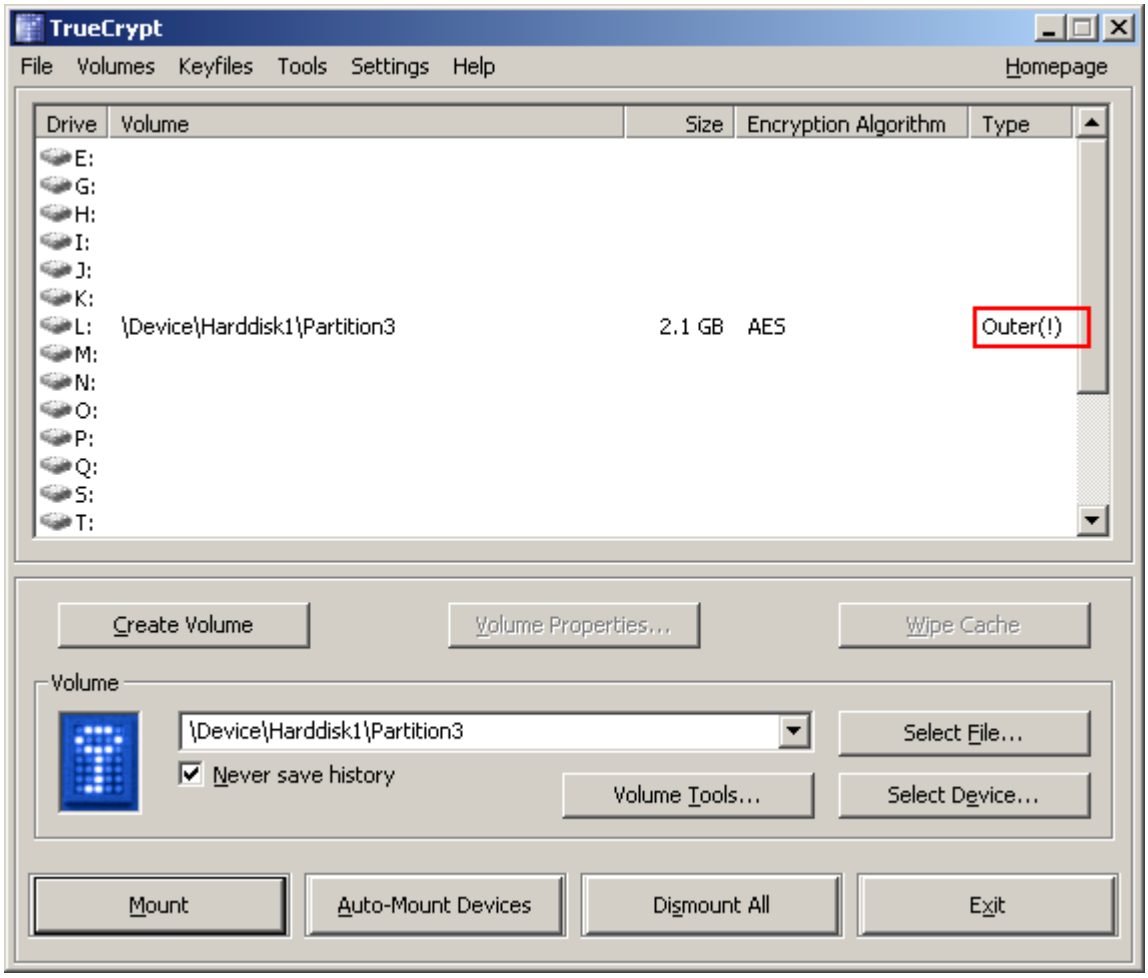
In the *Mount Options* dialog window, enable the option '*Protect hidden volume against damage caused by writing to outer volume* '. In the '*Password to hidden volume*' input field, type the password for the hidden volume. Click *OK* and in the main password entry dialog click *OK*.



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!)*'.

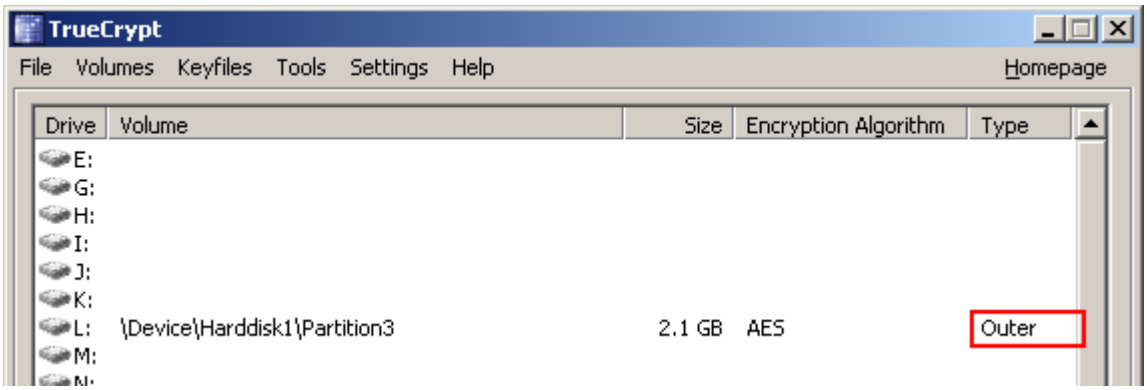
Note that when damage to hidden volume is prevented, *no* information about the event is written to the volume. When the outer volume is dismounted and mounted again, the volume properties will *not* display the string "*damage prevented*".

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!).

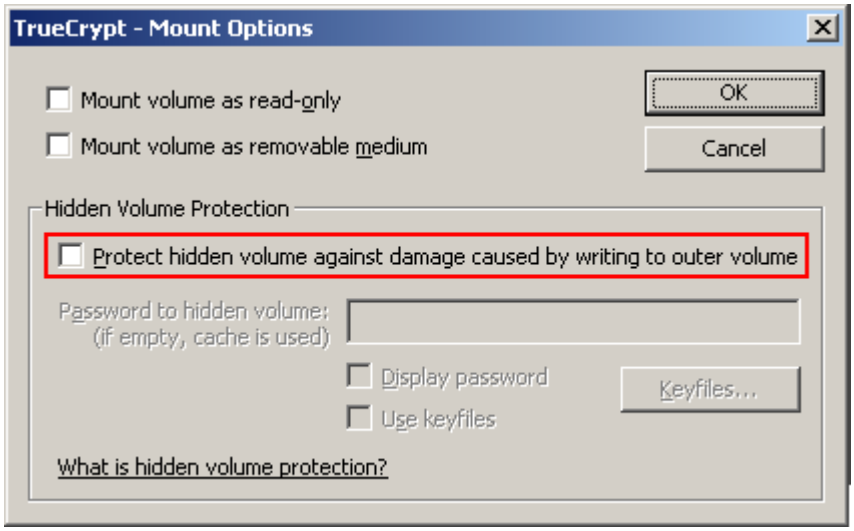
2. In the *Volume Properties* dialog, the field *Hidden Volume Protected* says '*Yes*':

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 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 cause the adversary to suspect that the volume contains a hidden 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.

- 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, 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).

TrueCrypt Volume

There are two types of TrueCrypt volumes:

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

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 (LRW 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 (LRW 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. Fore 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 ("greyed 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

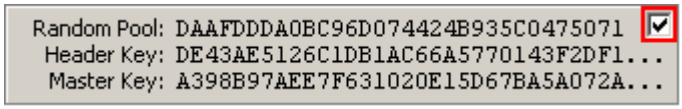
TrueCrypt supports hardware/software RAID as well as Windows dynamic volumes. If you intend to format a Windows dynamic volume as a TrueCrypt volume, keep in mind that after you create the Windows dynamic volume (using the Windows Disk Management tool), you must restart the operating system in order 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 (LRW 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).

Warning: When encrypting entire hard drive partition/device, i.e., formatting it as a TrueCrypt volume, all data stored on the partition/device will be lost!

Important: Several users reported that data on their TrueCrypt volumes were becoming corrupted. Later, these users found out that it was not a problem with TrueCrypt but with their hardware (chipset, USB hard drive, cables, USB PCI card, etc.) Therefore, we recommend that you make sure data written to the unencrypted device (where you intend to create a TrueCrypt volume) is not becoming corrupted. For example, by copying a large set of files (at least 5 GB in total) and then comparing the original files with the copies (by content).

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 which 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 traveller 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 ‘traveller’ mode, the TrueCrypt driver will be unloaded when it is no longer needed (e.g., when all instances of the main application and/or of the Volume Creation Wizard is closed and no TrueCrypt volumes are mounted).

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.

File -> Exit

Terminates the TrueCrypt application. The driver continues working and no TrueCrypt volumes are dismounted. When running in 'traveller' mode, the TrueCrypt driver will be unloaded when it is no longer needed (e.g., when all instances of the main application and/or of the Volume Creation Wizard is closed and no TrueCrypt volumes are mounted).

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 traveller 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*.

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-SHA-1 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 35 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunnelling microscopy [23] 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 (LRW 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*.

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 35 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunnelling microscopy [23] to recover the overwritten header (however, see also the chapter *Security Precautions*).

Tools -> Clear Volume History

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

Tools -> Traveller Disk Setup

See the chapter *Traveller Mode.*

Tools -> Keyfile Generator

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

Tools -> Backup Volume Header

If you do not have enough free space to backup all files stored on your TrueCrypt volume, we highly recommend that you at least backup the volume header (using this function), which contains the master key (size of the backup file will be 1024 bytes). If a volume header is damaged, the volume is, in most cases, impossible to mount.

To backup a volume header, click *Select Device* or *Select File* and select the volume. Then click *Tools -> Backup Volume Header*. To restore the header, follow the same steps except the last where you select *Restore Volume Header.*

A TrueCrypt volume header backup is just an exact copy of the encrypted volume header(s). The backup file does not contain any additional information. TrueCrypt volume header backups cannot be decrypted without knowing the correct password and/or supplying the correct keyfile(s).

Note that both the standard volume header and the area where hidden volume headers are stored will be backed up (copied to the backup file), even if there is no hidden volume within the volume (to preserve plausible deniability of hidden volumes). However, when *restoring* a volume header, you will choose which header should be restored (hidden or standard). 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*).

WARNING: Restoring a volume header also restores the volume password that was valid when the volume header 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.

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 that this facility 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*).

Tools -> Restore Volume Header

If a TrueCrypt volume becomes impossible to mount, it is possible that its header is corrupted. If you backed up the volume header, use this function to restore it.

When restoring a volume header, you have to choose which header is to be restored (a hidden or a standard volume header). Only one volume header can be restored at a time. To restore both headers, you need to use this function twice (*Tools -> Restore Volume Header*).

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.

Settings -> Preferences

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 traveller 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 created by the Recycle Bin and System Restore facilities).

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 various 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*.

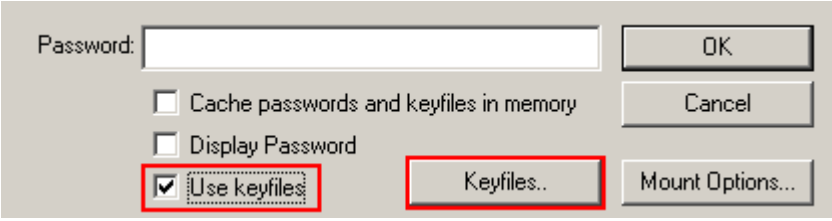
IMPORTANT: To make brute force attacks on a keyfile infeasible, the size of the keyfile should be at least 30 bytes. If a volume uses multiple keyfiles, then at least one of the keyfiles should be 30 bytes in size or larger. Note that the 30-byte limit assumes a large amount of entropy in the keyfile. 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).

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 button *Keyfile* 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 File*) *or* keyfile search paths (click *Add Path*). Note that keyfiles and keyfile search paths can also be selected by dragging the corresponding file/folder icons to the keyfile dialog window.

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 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 had used keyfiles from the folder will be impossible to mount (until you remove the newly added file from the folder).

* 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.

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.

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 35 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunnelling microscopy [23] 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 35 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunnelling microscopy [23] 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 that if you create/copy a new file in/to the folder, then all volumes that used the 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.*

Traveller Mode

TrueCrypt can run in so-called ‘traveller’ 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 ‘traveller’ 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 traveller 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 ‘traveller’ mode:

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

The second option has several advantages, which will be described in the following paragraphs.

Tools -> Traveller Disk Setup

You can use this facility to prepare a special ‘traveller’ disk and launch TrueCrypt from there. Note that TrueCrypt ‘traveller disk’ is *not* a TrueCrypt volume but an *unencrypted* volume. A ‘traveller disk’ contains TrueCrypt executable files and optionally the ‘autorun.inf’ script (see the section *AutoRun Configuration* below). After you select *Tools -> Traveller Disk Setup*, the *Traveller 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 ‘traveller’ disk you will create. Unchecking this option saves space on the ‘traveller’ disk.

AutoRun Configuration (autorun.inf)

In this section, you can configure the ‘traveller disk’ to automatically start TrueCrypt or mount a specified TrueCrypt volume when the ‘traveller disk’ is inserted. This is accomplished by creating a special script file called ‘*autorun.inf*’ on the traveller disk. This file is automatically executed by the operating system each time the ‘traveller 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 (or after the administrator gives the user administrator privileges). 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 'traveller' 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	LRW
Serpent	R. Anderson, E. Biham, L. Knudsen	256	128	LRW
Twofish	B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson	256	128	LRW
AES-Twofish		256; 256	128	LRW
AES-Twofish-Serpent		256; 256; 256	128	LRW
Serpent-AES		256; 256	128	LRW
Serpent-Twofish-AES		256; 256; 256	128	LRW
Twofish-Serpent		256; 256	128	LRW

For information about LRW 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 LRW mode (see the section *Modes of Operation*).

In June 2003, after the NSA (US National Security Agency) has 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 LRW 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 LRW 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 [24], 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 LRW mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with Twofish (256-bit key) and then with AES (256-bit key). 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 operating in LRW mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with Serpent (256-bit key), then with Twofish (256-bit key), and finally with AES (256-bit key). 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 operating in LRW mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with AES (256-bit key) and then with Serpent (256-bit key). 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 operating in LRW mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with AES (256-bit key), then with Twofish (256-bit key), and finally with Serpent (256-bit key). 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 operating in LRW mode (see the section *Modes of Operation*). Each 128-bit block is first encrypted with Serpent (256-bit key) and then with Twofish (256-bit key). 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 (LRW mode), and salt. For more information, please see the section *Random Number Generator* and section *Header Key Derivation, Salt, and Iteration Count*.

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 contest). 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].

SHA-1

SHA-1, published in 1995, is a hash algorithm designed by the NSA. The size of the output of this algorithm is 160 bits. In 2005, a theoretical method was published to find collisions in SHA-1 with effort smaller than that required for brute force on average (2^{63} instead of 2^{80} steps). However, as TrueCrypt uses SHA-1 merely as a pseudorandom function, it currently appears highly unlikely that possible future discovery of collisions in SHA-1 would affect the security of TrueCrypt volumes. This assumption is supported by proofs presented in [6]. Nevertheless, to be conservative, you may want to prefer Whirlpool or RIPEMD-160.

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, and represents a valuable alternative to SHA-1, which was designed by the NSA. The size of the output of RIPEMD-160 is 160 bits. RIPEMD-160 is a strengthened version of the RIPEMD hash algorithm which was developed in the framework of the European Union’s project RIPE (*RACE Integrity Primitives Evaluation*), 1988-1992, and in which collisions were found in 2004. No collisions have been found in RIPEMD-160 so far and no method is known to do so with effort smaller than that required for brute force on average (for information on how discovery of collisions in a hash function affects TrueCrypt, please see the section *SHA-1*). RIPEMD-160 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 2003
- Windows 2000
- Linux (kernel 2.6.5 or compatible)

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

Command Line Usage

Note that this section applies to the Windows version of TrueCrypt. Linux-specific features are described in the TrueCrypt man page, which is included in the TrueCrypt binary and source code distribution packages, which are available at: http://www.truecrypt.org/downloads.php.

/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`
/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. Consider using `/q` instead.*
/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. `/q background` launches the TrueCrypt Background Task (tray icon). Note that `/q` has no effect if the container is accessible only in local user name space (TrueCrypt will exit only after the volume is dismounted), e.g., a network volume.
/silent or */s*	If `/q` is specified, suppresses interaction with the user (prompts, error messages, warnings, etc.)
/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 volume/keyfile timestamps `persistent`: Do not display the volume in the GUI and prevent auto-dismount. '*Dismount All*' will not dismount the volume either. The volume can be dismounted only individually via command line. Note that this option works only if specified along with `/q`. `system`: Equal to `persistent` but in addition allows Windows paging files to be stored on the volume. Note that this option works only if specified along with `/q`. Example: `/m ro`. To specify multiple mount options, use e.g.: `/m rm /m ts`

Syntax

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

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
```

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

Also called ‘swap file’; Windows uses this file (usually stored on a hard disk) 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 disk 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).
Therefore, we strongly recommend that Windows XP/Vista users disable the paging file feature, at least for each session during which they work with sensitive data and during which they 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.*
To our best knowledge, Windows 2000 users cannot disable the paging file feature completely. We recommend that Windows 2000 users configure their Windows security settings to clear the paging files every time the system shuts down (refer to your Windows manual or www.microsoft.com for more information).

Hibernation Mode

When a computer hibernates (or enters a power-saving mode), the content of its system memory is written to a storage file on the hard drive. TrueCrypt *cannot* prevent cached passwords, encryption keys, and the contents of sensitive files opened in RAM from being saved *unencrypted* to a hibernation storage 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). Also note that when a TrueCrypt volume is mounted, its master key is stored *unencrypted* in RAM. Therefore, we strongly recommend that you prevent or disable hibernation mode on your computer at least for each session during which your work with any sensitive data and during which you mount a TrueCrypt volume.

Memory Dump Files

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 computer at least for each session during which your 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.*

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).

Unencrypted Data in RAM

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. 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.

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.

Secondary Key

When an LRW tweak key that is part of the master key set with which a TrueCrypt volume is encrypted is saved to the same TrueCrypt volume, the attacker may be able to recover the tweak key (even when the volume is not mounted). This might happen, for example, when you mount a TrueCrypt volume and then save the part of RAM that holds the tweak key of the volume to a file on the volume (see also sections *Memory Dump Files*, *Paging File*, and *Hibernation Mode*).

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. 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).

Data Corruption

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).

If you do not have enough free space to backup all files, we highly recommend that you at least backup the volume header, which contains the master key (size of the backup file will be 1024 bytes). If a volume header is damaged, the volume is, in most cases, impossible to mount. To backup a volume header, click *Select Device* or *Select File* and select the volume. Then click *Tools -> Backup Volume Header*. To restore the header, follow the same steps except the last where you select *Restore Volume Header.*

Important: Several users reported that data on their TrueCrypt volumes were becoming corrupted. Later, these users found out that it was not a problem with TrueCrypt but with their hardware (chipset, USB hard drive, cables, USB PCI card, etc.) Therefore, we recommend that you make sure data written to the unencrypted device (where you intend to create a TrueCrypt volume) is not becoming corrupted. For example, by copying a large set of files (at least 1 GB in total) and then comparing the original files with the copies (by content) using a file comparison utility.

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.

---

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

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 stored on devices (or in file systems) that utilize a wear-leveling mechanism. To find out whether a device utilizes a wear-leveling mechanism, please refer to documentation supplied with the device or contact the vendor/manufacturer.

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). 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).

See also the section *Security Precautions Pertaining to Hidden Volumes*.

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:

*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:

*After successfully mounting a volume, it cannot be accessed via Windows Explorer (it is not visible in the '*Computer*' or '*My Computer*' list, etc.) even though the volume is displayed in TrueCrypt as mounted. Sometimes Windows Explorer does not display the correct volume label.*

PROBABLE CAUSE:

A Windows Explorer issue.

POSSIBLE SOLUTION:

Click *Tools -> Refresh Drive Letters*. If it does not help, restart Windows Explorer (for example, by logging off and on). Note that you can also open a volume by double clicking its drive letter in the drive list in the main TrueCrypt window.

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:

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:

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. Should you be asked why the volume has such a large cluster size, you can answer that you prefer higher performance (see the section *Cluster Size* for more information).

PROBLEM:

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

POSSIBLE SOLUTION:

First, make sure that you are not trying to encrypt the operating system boot partition (TrueCrypt does not support this). Then 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:

1. *A TrueCrypt volume cannot be mounted*
2. *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 network, “insufficient memory” 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/default.aspx?scid=kb;en-us;285089

Incompatibilities

There are currently no confirmed incompatibilities.

Known Issues & Limitations

- 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 not supported.

- TrueCrypt-encrypted floppy disks: When a floppy disk is ejected and another one is inserted, garbage will be read/written to the disk, which could lead to data corruption. 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: *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 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: *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, it is. For more information, please visit http://www.truecrypt.org/donations/*

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* Traveller Mode*).*

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

A: *TrueCrypt volumes can be up to 8589934592 GB. However, you need to take into account several limiting factors. For instance, file system constraints, limitations of the hardware connection standard and of the operating system, etc.*

Q: *I've heard that SHA-1 has been broken. Does it affect TrueCrypt?*

A: *SHA-1 is one of the three hash algorithms used by TrueCrypt (the user decides which algorithm is used). In 2005, a theoretical method was invented to find collisions in SHA-1 with effort smaller than that required for brute force on average (2^{63} instead of 2^{80} steps). However, as TrueCrypt uses SHA-1 merely as a pseudorandom function, it currently appears to be highly unlikely that possible future discovery of collisions in SHA-1 would affect the security of TrueCrypt volumes. This assumption is supported by proofs presented in M. Bellare's paper* New Proofs for NMAC and HMAC: Security without Collision-Resistance*. Nevertheless, to be conservative, you may want to prefer Whirlpool or RIPEMD-160. For more information, please see the chapter* Hash Algorithms*.*

Q: *Is TrueCrypt distributed under an open source license such as the GPL?*

A: *Yes, it is (however, not under the GPL). 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/license.php.*

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

A: *File containers can be easily moved, renamed, and managed like normal files (however, this also means that a container may get damaged or deleted as easy as any other file). Partitions/devices may be better as regards performance. Note that reading to/writing from a file container may take significantly longer when the container is heavily fragmented. Also note that mounting a hidden volume located within a file container may take significantly longer when the container is heavily fragmented. The reason is that the header of the hidden volume is located at the end of the outer (host) container and seeking the end of the container may take a long time when the container is fragmented. To solve this problem, defragment the container (when it is dismounted).*

Q: *Will I be able to mount my TrueCrypt partition/container on any computer?*

A: *TrueCrypt volumes 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: *Will I be able to mount my TrueCrypt partition/container after I reinstall 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.*

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 sector of the volume where hidden volume headers are normally stored (the third sector from the end of the volume) to RAM and attempts to decrypt it using the entered password. Note that 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).

Further information may be found in the section Hidden Volume.

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

A: *Yes, but only after a system administrator installs TrueCrypt on the system (or after he or she gives you administrator privileges). 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 'traveller' mode.*

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

A: *Yes, it does. 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: How does TrueCrypt verify that the correct password was entered?

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

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

A: *Yes, it is. 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 NTFS file system on read-only media).*

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

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

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

A: *When you run TrueCrypt in traveller 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 traveller mode), you will not be asked for permission every time you run it.

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: *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 following options:*
    - *'*Start TrueCrypt*'*
    - *'*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 or HMAC-SHA-1, 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, convert it from HMAC-SHA-1 to HMAC-Whirlpool) without losing data stored on the volume?*

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

Q: *Can the latest version of TrueCrypt mount volumes encrypted in CBC mode (i.e., volumes created by TrueCrypt 4.0 or earlier)?*

A: *Yes, it can. However, note that LRW mode is more secure than CBC mode. Therefore, we strongly recommend you to create a new volume move using the latest version of TrueCrypt and move data from your old volume to it. CBC mode has been deprecated and is only supported as legacy.*

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: *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 without losing the data currently stored on it?*

A: *No, TrueCrypt does not allow this, and we do not plan to implement such feature either (there are several reasons for our decision and most of them are security-related).*

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 Windows?*

A: *Yes. This can be achieved by running TrueCrypt in traveller 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 disk 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. Simply boot BartPE, download the latest version of TrueCrypt to the RAM disk (which BartPE creates), extract the downloaded archive to the RAM disk, and run the file 'TrueCrypt.exe' from the folder 'Setup Files' on the RAM disk (the 'Setup Files' folder should be created when you unpack the archive containing TrueCrypt).*

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, it does. 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 Windows XP x64 (64-bit) Edition?*

A: *Yes, it does.*

Q: *Does TrueCrypt run on Windows 98 or Windows ME?*

A: *The last version of TrueCrypt that ran on Windows 98/ME was 1.0. Note that we do not support TrueCrypt 1.0 and do not recommend using it (see the section* Version History *for more information).*

Q: *Does TrueCrypt run on Linux?*

A: *Yes.*

Q: *Can I mount my TrueCrypt volume both under Windows and under 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*).*

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). If you do not have enough free space to backup all files, we highly recommend that you at least backup the volume header, which contains the master key (size of the backup file will be 1024 bytes). If a volume header is damaged, the volume is, in most cases, impossible to mount. To backup a volume header, click Select Device *or* Select File *and select the volume. Then click* Tools -> Backup Volume Header*. To restore the header, follow the same steps except the last where you select* Restore Volume Header*.*

*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 environment. Is there a way for an administrator to reset a volume password when a user forgets it (or when he or she loses the keyfile)?*

A: *There is no "back door" implemented in TrueCrypt. However, there* is *a way to "reset" a TrueCrypt volume password/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 (*Tools -> Restore Volume Header*).*

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

A: Yes, but the volume must be mounted in read-only mode under each of the systems (see the section Mount Options*). 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).*

Q: *How do I decrypt a TrueCrypt volume permanently?*

A: *Please note that TrueCrypt does not support in-place decryption. If you no longer wish to use TrueCrypt, 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, right-click the drive letter of the partition in the '*Computer*' (or '*My Computer*') window and select* Format *(to open the '*Computer*' window, click the '*Computer*' icon in the* Start Menu *or double-click it on your desktop). The Format window should appear. WARNING: Before you continue, make sure you have selected the correct drive letter. In the* Format *window, click* Start*. This will format the partition and 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. *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 either* Format *(if it is an unpartitionable removable medium, such as a USB flash drive) or* New Partition *(if it is a partitionable medium).*
   d. *Depending on what you selected in the previous step, either follow the instructions provided by the '*New Partition Wizard*' or click* OK *in the* Format *window.*

Q: *Will I always be able to mount a TrueCrypt container no matter how fragmented it is?*

A: *Yes. However, note that reading to/writing from a file container may take significantly longer when the container is heavily fragmented. Also note that mounting a hidden volume located within a file container may take significantly longer when the container is heavily fragmented. The reason is that the header of the hidden volume is located at the end of the outer (host) container and seeking the end of the container may take a long time when the container is fragmented. To solve this, defragment the whole host container (when it is dismounted) or create a hidden volume within a partition or a device.*

Q: *Is it necessary to restart the computer before copying a TrueCrypt container?*

A: *No, it is not necessary.*

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 automatically created 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: *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: *Why is it not possible to create arbitrary cascades of ciphers?*

A: *The reason is that the encryption algorithm (and the mode of operation) that a TrueCrypt volume has been encrypted with is unknown. The correct encryption algorithm has to be determined through the process of trial and error. If we added the support for creating arbitrary cascades, the number of encryption algorithms to attempt mounting with would increase tremendously. The time needed to mount a volume would no longer be acceptable especially on slow computers.*

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).*

Q: Can TrueCrypt encrypt a Windows boot partition?

A: Yes, but not directly. TrueCrypt can on-the-fly encrypt a disk image containing an installed operating system that you run (boot) under virtual machine (or emulation) software, such as Virtual PC, VirtualBox, VMware, QEMU, or Bochs (Bochs, QEMU, and VirtualBox OSE are free and open-source; Virtual PC and some editions of VMware are free).

Uninstalling TrueCrypt

To uninstall TrueCrypt, open the Windows Control Panel and select '*Add/Remove Programs*'. Locate TrueCrypt and click the '*Add/Remove*' button.

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 'traveller' 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`
Note: This file is not present if TrueCrypt is run in 'traveller' 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 traveller 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 these files.**

`Configuration.xml`

`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
$\wedge$	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: $1 \oplus 0 = 1$; $1 \oplus 1 = 0$; $1 \oplus 2 = 1$; $1 \oplus 3 = 0$; $0 \oplus 0 = 0$; $0 \oplus 1 = 1$; $0 \oplus 2 = 0$; $0 \oplus 3 = 1$)
$\otimes$	Multiplication in the finite field $GF(2^{128})$; multiplication of two polynomials modulo $x^{128}+x^7+x^2+x+1$ (reduction polynomial)
$\|\|$	Concatenation

Encryption Scheme

When mounting a TrueCrypt volume (assume there are no cached passwords/keyfiles), 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*).

2. The 512 bytes at byte #1536 (offset) from the end of the volume are read into RAM (see the section *TrueCrypt Volume Format Specification*). If there is a hidden volume within this volume, at this point we have read its header (whether or not there is a hidden volume within this volume 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-RIPEMD-160, HMAC-SHA-1, 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 (LRW 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: LRW, CBC (*deprecated/legacy*), inner-CBC (*deprecated/legacy*), outer-CBC (*deprecated/legacy*)

    d. Key size(s)

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

[*] These parameters are kept secret not in order to increase the complexity of an attack, but primarily to make TrueCrypt volumes unidentifiable (undistinguishable from random data), which would be difficult to achieve if these parameters were stored within the volume header.

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 reinitialised with the master key[*] and secondary key (LRW mode), 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 (which has been encrypted using the header keys). The volume is mounted.

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

Modes of Operation

Volumes created by this version of TrueCrypt can be encrypted only in LRW mode. CBC mode has been deprecated (however, volumes encrypted in CBC mode can still be mounted by the current version of TrueCrypt). LRW mode is more secure than CBC mode and is suitable for disk encryption.

Description of LRW mode:

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

Where:
$K1$ is the encryption key
$K2$ is the secondary key (sometimes referred to as "tweak" key)
i is the cipher block index within the scope of $K1$; for the first cipher block, $i = 1$
$\otimes$ denotes multiplication of two polynomials modulo $x^{128}+x^7+x^2+x+1$

$K2$ and i are 128-bit values.

For further information pertaining to LRW mode, see e.g. [12].

The following table lists all encryption algorithms implemented in TrueCrypt and the modes in which they operate:

[*] The master key was 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).

Encryption Algorithm	Mode of Operation	Details of Mode of Operation
AES*	LRW	$C_i = E_{K1}(P_i \wedge (K2 \otimes i)) \wedge (K2 \otimes i)$
AES-Twofish ($E2$) ($E1$)	LRW	$C_i = E2_{K2}(E1_{K1}(P_i \wedge (K3 \otimes i))) \wedge (K3 \otimes i)$
AES-Twofish-Serpent ($E3$) ($E2$) ($E1$)	LRW	$C_i = E3_{K3}(E2_{K2}(E1_{K1}(P_i \wedge (K4 \otimes i)))) \wedge (K4 \otimes i)$
Serpent	LRW	$C_i = E_{K1}(P_i \wedge (K2 \otimes i)) \wedge (K2 \otimes i)$
Serpent-AES ($E2$) ($E1$)	LRW	$C_i = E2_{K2}(E1_{K1}(P_i \wedge (K3 \otimes i))) \wedge (K3 \otimes i)$
Serpent-Twofish-AES ($E3$) ($E2$) ($E1$)	LRW	$C_i = E3_{K3}(E2_{K2}(E1_{K1}(P_i \wedge (K4 \otimes i)))) \wedge (K4 \otimes i)$
Twofish	LRW	$C_i = E_{K1}(P_i \wedge (K2 \otimes i)) \wedge (K2 \otimes i)$
Twofish-Serpent ($E2$) ($E1$)	LRW	$C_i = E2_{K2}(E1_{K1}(P_i \wedge (K3 \otimes i))) \wedge (K3 \otimes i)$

Ciphers in a cascade use mutually independent keys (note that the header keys they use are independent as well, even though they are derived from a single password – see the section *Header Key Derivation, Salt, and Iteration Count*).

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 (LRW 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 significantly 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

* In this table, we use the term "AES" to refer to "AES-256". AES operating in LRW mode is also referred to as LRW-AES.

derivation function is based on either HMAC-RIPEMD-160, or HMAC-Whirlpool, or HMAC-SHA-1 (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 (e.g.., header key for the AES-256 cipher is always 256 bits long, even if HMAC-SHA-1 or HMAC-RIPEMD-160 is used). For more information, refer to [7]. Two thousand iterations (or one thousand iterations when HMAC-Whirlpool 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 key from a given password. The generated key is then split into three 256-bit keys, out of which the first key is used by Serpent, the second key is used by Twofish, and the third 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 (LRW mode), salt, and keyfiles. It creates a pool of random values in RAM (memory). The pool, which is 320 bytes long, is filled with data from the following sources:

- Mouse movements
- Keystrokes*
- *Linux only*: Values generated by the Linux 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 eighth byte written to the pool, the pool mixing function is applied to the entire pool (see below).

* Under Linux, keystrokes are read only when no mouse is available.

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 eighth 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 (RIPEMD-160, SHA-1, or Whirlpool)
3. l = byte size of the output of the hash function H (i.e., if H is SHA-1 or RIPEMD-160, then $l = 20$; if H is Whirlpool, $l = 64$)
4. z = byte size of the randomness pool R (320 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. There are no forced restrictions on the contents of a keyfile. The user can 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 (number of keyfiles is not limited).

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 (LRW mode).

TrueCrypt Volume Format Specification

The format of file-hosted volumes is identical to the format of partition/device-hosted volumes. TrueCrypt volume has no “signature“ or ID string. Until decrypted, it appears to consist of random data entirely. Therefore, it is impossible to identify a TrueCrypt container or partition.

Free space of each TrueCrypt volume is filled with random data when the volume is created (if the options *Quick Format* and *Dynamic* are disabled). The random data is generated as follows: Right before TrueCrypt volume formatting begins, a temporary encryption key and a temporary secondary key (LRW mode) are generated by the random number generator (see the section *Random Number Generator*). The encryption algorithm that the user selected is initialised 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 LRW 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.

TrueCrypt Volume Format Specification:

Offset (bytes)	Size (bytes)	Encryption Status*	Description
0	64	Not Encrypted†	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	8	Encrypted	Volume creation time
84	8	Encrypted	Header creation/modification time
92	8	Encrypted	Reserved (set to zero)
100	156	Encrypted	Currently unused
256	*Var.*	Encrypted	Secondary master key (LRW mode)
288	*Var.*	Encrypted	Master key(s)‡
512	*Var.*	Encrypted	Data area (actual volume contents)

The fields located at the byte #0 (salt), at the byte #256 (secondary key), and at the byte #288 (master encryption keys), contain random values that have been generated by the random number generator (see the section *Random Number Generator*) during the volume creation process.

* The encrypted areas of the volume header are encrypted with the header key (and the secondary header key in LRW mode). For more information, see the section *Encryption Scheme* and the section *Header Key Derivation, Salt, and Iteration Count*.
† 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 master keys are stored here when the volume is encrypted using a cascade of ciphers.

If a TrueCrypt volume hosts a hidden volume (within its free space), the header of the hidden volume is located at the byte #1536 (offset) from the end of the host volume (the header of the host/outer volume is located at the beginning of the volume – see the section *Hidden Volume*). The format of the hidden volume header is specified in the following table:

Offset (bytes)	Size (bytes)	Encryption Status	Description
0	64	Not Encrypted	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	8	Encrypted	Volume creation time
84	8	Encrypted	Header creation/modification time
92	8	Encrypted	Size of the hidden volume
100	156	Encrypted	Currently unused
256	*Var.*	Encrypted	Secondary master key (LRW mode)
288	*Var.*	Encrypted	Master key(s)

The size of a TrueCrypt volume header is always 512 bytes. The size of the hidden volume header is also 512 bytes.

The maximum supported TrueCrypt volume size is 8,589,934,592 GB (i.e., 2^{63} bytes).

Compliance with Standards and Specifications

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

- PKCS #5 v2.0 [7]
- FIPS 46-3 [13]
- 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 (click *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 the C and C++ programming languages) is freely available for peer review at:

http://www.truecrypt.org/downloads.php

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/license.php

Contact

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

Version History

4.3

March 19, 2007

New features:

- Full compatibility with 32-bit and 64-bit Windows Vista:
    - Support for User Account Control (UAC).
    - All .sys and .exe files of TrueCrypt are now digitally signed with the digital certificate of the TrueCrypt Foundation, which was issued by the certification authority GlobalSign.
    - When moving the mouse on a single-CPU computer while reading or writing data to a TrueCrypt volume, the mouse pointer stopped moving for a second every few seconds. This will no longer occur. (*Windows Vista issue*)
- TrueCrypt volume is automatically dismounted if its host device is inadvertently removed.

  Important: You should always dismount the volume in TrueCrypt and then use the "*Safely Remove Hardware*" function (built in Windows) before you physically remove the host device (e.g. a USB flash drive).
- Support for devices and file systems that use a sector size other than 512 bytes (e.g., new hard drives, USB flash drives, DVD-RAM, MP3 players, etc.)
- Support for devices with a GPT partition table (GUID partitions). (*Windows Vista/2003/XP*)
- After a partition is successfully encrypted, the drive letter assigned to it (if any) is automatically removed. (*Windows*)
- Volume name (label) is displayed in device/partition selector. (*Windows*)
- New hotkey: 'Wipe Cache'. (*Windows*)
- New command line switch '*/q background*' for launching the TrueCrypt Background Task. (*Windows*)

Improvements:

- Portions of the TrueCrypt device driver redesigned.
- Maximum allowed size of FAT32 volumes increased to 2 TB (note that NTFS volumes can be larger than 2 TB).
- Traveller Disk Setup improved. (*Windows*)
- Volumes hosted on read-only media will always be mounted in read-only mode. (*Windows*)
- Improved support for big-endian platforms.
- Other minor improvements (*Windows and Linux*)

Bug fixes:

- The built-in FAT format facility now functions correctly on big-endian platforms.

- Improved handling of partitions and devices during volume creation. (*Windows*)
- Improved handling of low-memory conditions. (*Windows*)
- Fixed bug that rarely caused system errors when dismounting all volumes. (*Windows*)
- Tray icon is recreated when Windows Explorer is restarted (e.g. after a system crash).
- Other minor bug fixes (*Windows and Linux*)

Security improvements:

- Improved security of set-euid mode of execution. A volume can be dismounted only by the user who mounted it or by an administrator (root). (*Linux*)

Removed features:

- It is no longer possible to create new volumes encrypted with 64-bit-block encryption algorithms (Blowfish, CAST-128, and Triple DES). 64-bit block ciphers are being phased out. It is still possible to mount such volumes using this version of TrueCrypt. However, it will not be possible to mount such volumes using TrueCrypt 5.0 and later versions (this applies also to volumes encrypted with AES-Blowfish and AES-Blowfish-Serpent, which have been in the process of being phased out since TrueCrypt 4.1). If you have such a volume, we recommend that you create a new TrueCrypt volume encrypted with a 128-bit-block encryption algorithm (e.g., AES, Serpent, Twofish, etc.) and that you move files from the old volume to the new one.

For a list of changes in older version, please refer to 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; TrueCrypt is based on E4M.

Dr. Brian Gladman, who wrote the excellent AES, Twofish, SHA-1, and finite field GF(2^{128}) multiplication routines.

Eric Young, who wrote the excellent Triple-DES, Blowfish, and CAST5 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 routines, and *Dag Arne Osvik* for his paper *Speeding up Serpent*.

Markus Friedl, who wrote the RIPEMD-160 routines (taken from OpenBSD).

The designers of the encryption and hash algorithms:
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, 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://csrc.nist.gov/CryptoToolkit/aes/round2/r2report.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*, Request for Comments 2104, February 1997, available at http://www.ietf.org/rfc/rfc2104.txt.

[9] P. Cheng, IBM, R. Glenn, NIST, *Test Cases for HMAC-MD5 and HMAC-SHA-1*, Request for Comments 2202, February 1997, available at http://www.ietf.org/rfc/rfc2202.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] M. Liskov, R. Rivest, D. Wagner, *Tweakable Block Ciphers*, Advances in Cryptology – CRYPTO '02, vol. 2442 of Lecture Notes in Computer Science, pp. 31-46. Springer-Verlag, 2002; also available at: http://theory.lcs.mit.edu/~rivest/LiskovRivestWagner-TweakableBlockCiphers.pdf

[13] NIST, *Data Encryption Standard*, Federal Information Processing Standards Publication 46-3, October 25, 1999, available at http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

[14] NIST, *Secure Hash Standard*, 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] List of the approved cryptographic algorithms for the protection of Protected Information within the Government of Canada: http://www.cse-cst.gc.ca/en/services/crypto_services/crypto_algorithms-e.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] 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

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