This documentation is not guaranteed to be error-free and is provided "as is" without warranty of any kind. For more information, see Disclaimers.

Introduction

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

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

Let's suppose that there is an .avi video file stored on a TrueCrypt volume (therefore, the video file is entirely encrypted). The user provides the correct password (and/or keyfile) and mounts (opens) the TrueCrypt volume. When the user double clicks the icon of the video file, the operating system launches the application associated with the file type – typically a media player. The media player then begins loading a small initial portion of the video file from the TrueCrypt-encrypted volume to RAM (memory) in order to play it. While the portion is being loaded, TrueCrypt is automatically decrypting it (in RAM). The decrypted portion of the video (stored in RAM) is then played by the media player. While this portion is being played, the media player begins loading another 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).

For a quick start guide, please see the chapter **Beginner's Tutorial**.

Beginner's Tutorial

How to Create and Use a TrueCrypt Container

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

Step 1:

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

Step 2:

System Encryption

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

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

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

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

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

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

Note: By default, Windows 7 and later boot from a special small partition. The partition contains files that are required to boot the system. Windows allows only applications that have administrator privileges to write to the partition (when the system is running). TrueCrypt encrypts the partition only if you choose to encrypt the whole system drive (as opposed to choosing to encrypt only the partition where Windows is installed).

Operating Systems Supported for System Encryption

TrueCrypt can currently encrypt the following operating systems:

- Windows 7 (32-bit and 64-bit)
- Windows Vista (SP1 or later)
- Windows Vista x64 (64-bit) Edition (SP1 or later)
- Windows XP
- Windows XP x64 (64-bit) Edition
- Windows Server 2008 R2 (64-bit)
- Windows Server 2008
- Windows Server 2008 x64 (64-bit)
- Windows Server 2003
- Windows Server 2003 x64 (64-bit)

Note: The following operating systems (among others) are not supported: Windows RT, Windows 2003 IA-64, Windows 2008 IA-64, Windows XP IA-64, and the Embedded/Tablet versions of Windows.

See also: Supported Operating Systems

Next Section >>

Hidden Operating System

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

Next Section >>

TrueCrypt Rescue Disk

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

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

- If the **TrueCrypt Boot Loader is frequently damaged** (for example, by inappropriately designed activation software) or if **you do not want the TrueCrypt boot loader to reside on the hard drive** (for example, if you want to use an alternative boot loader/manager for other operating systems), you can boot directly from the TrueCrypt Rescue Disk (as it contains the TrueCrypt boot loader too) without restoring the boot loader to the hard drive. Just insert your Rescue Disk into your CD/DVD drive and then enter your password in the Rescue Disk screen.

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

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

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

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

Plausible Deniability

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

1. Hidden volumes (see the section **Hidden Volume**) and hidden operating systems (see the section **Hidden Operating System**).

2. Until decrypted, a TrueCrypt partition/device appears to consist of nothing more than random data (it does not contain any kind of "signature"). Therefore, it should be impossible to prove that a partition or a device is a TrueCrypt volume or that it has been encrypted (provided that the security requirements and precautions listed in the chapter Security Requirements and Precautions are followed). A possible plausible explanation for the existence of a partition/device containing solely random data is that you have wiped (securely erased) the content of the partition/device using one of the tools that erase data by overwriting it with random data (in fact, TrueCrypt can be used to securely erase a partition/device too, by creating an empty encrypted partition/device-hosted volume within it). However, you need to prevent data leaks (see the section Data Leaks) and also note that, for system encryption, the first drive track contains the (unencrypted) TrueCrypt Boot Loader, which can be easily identified as such (for more information, see the chapter System Encryption). When using system encryption, plausible deniability can be achieved by creating a hidden operating system (see the section Hidden Operating System).

   Although file-hosted TrueCrypt volumes (containers) do not contain any kind of "signature" either (until decrypted, they appear to consist solely of random data), they cannot provide this kind of plausible deniability, because there is practically no plausible explanation for the existence of a file containing solely random data. However, plausible deniability can still be achieved with a file-hosted TrueCrypt volume (container) by creating a hidden volume within it (see above).

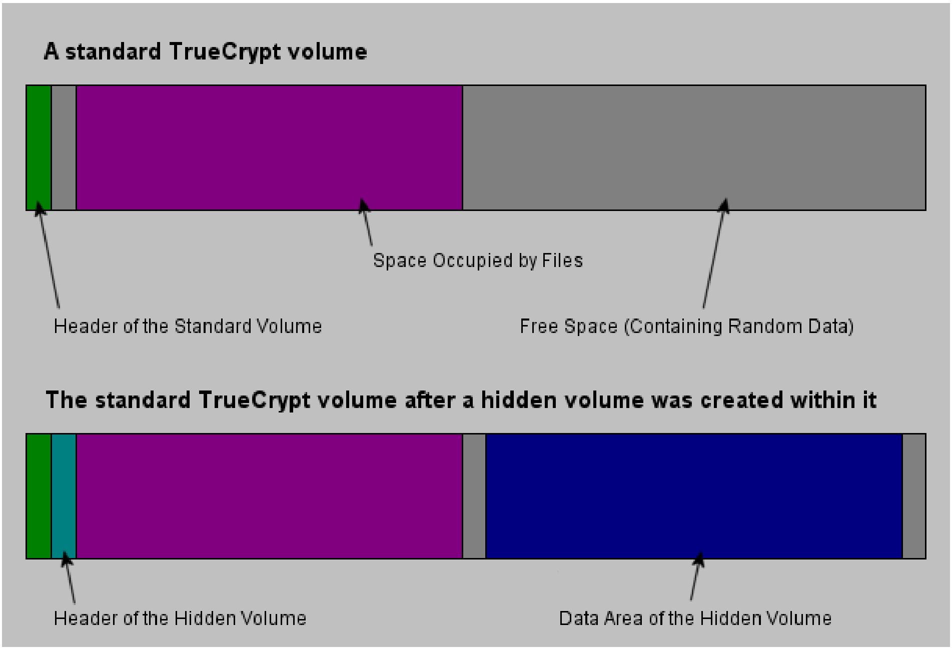
Notes

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

- There are methods to find files or devices containing random data (such as TrueCrypt volumes). Note, however, that this should *not* affect plausible deniability in any way. The adversary still should not be able to *prove* that the partition/device is a TrueCrypt volume or that the file, partition, or device, contains a hidden TrueCrypt volume (provided that you follow the security requirements and precautions listed in the chapter Security Requirements and Precautions and in the subsection Security Requirements and Precautions Pertaining to Hidden Volumes).

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.


A standard TrueCrypt volume
Space Occupied by Files
Header of the Standard Volume
Free Space (Containing Random Data)
The standard TrueCrypt volume after a hidden volume was created within it
Header of the Hidden Volume
Data Area of the Hidden Volume

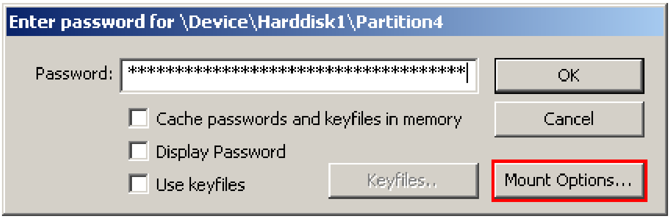
The layout of a standard TrueCrypt volume before and after a hidden volume was created within it.

The principle is that a TrueCrypt volume is created within another TrueCrypt volume (within the f

Protection of Hidden Volumes Against Damage

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

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


Enter password for \Device\Harddisk1\Partition4
Password: ***************************************
OK
Cancel
□ Cache passwords and keyfiles in memory
□ Display Password
□ Use keyfiles
Keyfiles...
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*.

Security Requirements and Precautions Pertaining to Hidden Volumes

If you use a hidden TrueCrypt volume, you must follow the security requirements and precautions listed below in this section. Disclaimer: This section is not guaranteed to contain a list of *all* security issues and attacks that might adversely affect or limit the ability of TrueCrypt to secure data stored in a hidden TrueCrypt volume and the ability to provide plausible deniability.

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

  Note that issues similar to the one 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/filesystem that utilizes a wear-leveling mechanism (e.g. a flash-memory SSD or USB flash drive). A copy of (a fragment of) the TrueCrypt volume may remain on the device. Therefore, do not store hidden volumes on such devices/filesystems. For more information on wear-leveling, see the section Wear-Leveling in the chapter Security Requirements and Precautions.

  - A TrueCrypt volume resides on a device/filesystem that saves data (or on a device/filesystem that is controlled or monitored by a system/device that saves data) (e.g. the value of a timer

Hidden Operating System

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

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

A **hidden operating system** is a system (for example, Windows 7 or Windows XP) that is installed in a hidden TrueCrypt volume. It should be impossible to prove that a hidden TrueCrypt volume exists (provided that certain guidelines are followed; for more information, see the section Hidden Volume) and, therefore, it should be impossible to prove that a hidden operating system exists.

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

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

There will be two pre-boot authentication passwords — one for the hidden system and the other for the decoy system. If you want to start the hidden system, you simply enter the password for the hidden

Parallelization

When your computer has a multi-core processor (or multiple processors), TrueCrypt uses all of the cores (or processors) in parallel for encryption and decryption. For example, when TrueCrypt is to decrypt a chunk of data, it first splits the chunk into several smaller pieces. The number of the pieces is equal to the number of the cores (or processors). Then, all of the pieces are decrypted in parallel (piece 1 is decrypted by thread 1, piece 2 is decrypted by thread 2, etc). The same method is used for encryption.

So if your computer has, for example, a quad-core processor, then encryption and decryption are four times faster than on a single-core processor with equivalent specifications (likewise, they are twice faster on dual-core processors, etc).

Increase in encryption/decryption speed is directly proportional to the number of cores and/or processors.

Note: Processors with the Hyper-Threading technology provide multiple logical cores per one physical core (or multiple logical processors per one physical processor). When Hyper Threading is enabled in the computer firmware (e.g. BIOS) settings, TrueCrypt creates one thread for each logical core/processor. For example, on a 6-core processor that provides two logical cores per one physical core, TrueCrypt uses 12 threads.

When your computer has a multi-core processor/CPU (or multiple processors/CPUs), header key derivation is parallelized too. As a result, mounting of a volume is several times faster on a multi-core processor (or multi-processor computer) than on a single-core processor (or a single-processor computer) with equivalent specifications.

Note: Parallelization was introduced in TrueCrypt 6.0.

See also: **Pipelining**, **Hardware Acceleration**

Pipelining

When encrypting or decrypting data, TrueCrypt uses so-called pipelining (asynchronous processing). While an application is loading a portion of a file from a TrueCrypt-encrypted volume/drive, TrueCrypt is automatically decrypting it (in RAM). Thanks to pipelining, the application does not have wait for any portion of the file to be decrypted and it can start loading other portions of the file right away. The same applies to encryption when writing data to an encrypted volume/drive.

Pipelining allows data to be read from and written to an encrypted drive as fast as if the drive was not encrypted (the same applies to file-hosted and partition-hosted TrueCrypt volumes).*

Note: Pipelining was introduced in TrueCrypt 5.0 and it is implemented only in the Windows versions of TrueCrypt.

* Some solid-state drives compress data internally, which appears to increase the actual read/write speed when the data is compressible (for example, text files). However, encrypted data cannot be compressed (as it appears to consist solely of random "noise" without any compressible patterns). This may have various implications. For example, benchmarking software that reads or writes compressible data (such as sequences of zeroes) will report lower speeds on encrypted volumes than on unencrypted volumes (to avoid this, use benchmarking software that reads/writes random or other kinds of uncompressible data).

See also: **Parallelization**, **Hardware Acceleration**

Hardware Acceleration

Some processors (CPUs) support hardware-accelerated AES encryption,* which is typically 4-8 times faster than encryption performed by the purely software implementation on the same processors.

By default, TrueCrypt uses hardware-accelerated AES on computers that have a processor where the Intel AES-NI instructions are available. Specifically, TrueCrypt uses the AES-NI instructions that perform so-called AES rounds (i.e. the main portions of the AES algorithm).** TrueCrypt does not use any of the AES-NI instructions that perform key generation.

Note: By default, TrueCrypt uses hardware-accelerated AES also when an encrypted Windows system is booting or resuming from hibernation (provided that the processor supports the Intel AES-NI instructions).

To find out whether TrueCrypt can use hardware-accelerated AES on your computer, select *Settings > Performance* and check the field labeled '*Processor (CPU) in this computer supports hardware acceleration for AES*'.

To find out whether a processor you want to purchase supports the Intel AES-NI instructions (also called "AES New Instructions"), which TrueCrypt uses for hardware-accelerated AES, please check the documentation for the processor or contact the vendor/manufacturer. Alternatively, peruse this official list of Intel processors that support the AES-NI instructions or use the official AMD website to find such AMD processors. However, note that some Intel processors, which the Intel website lists as AES-NI-supporting, actually support the AES-NI instructions only with a Processor Configuration update (for example, i7-2630/2635QM, i7-2670/2675QM, i5-2430/2435M, i5-2410/2415M). In such cases, you should contact the manufacturer of the motherboard/computer for a BIOS update that includes the latest Processor Configuration update for the processor.

If you want to disable hardware acceleration of AES (e.g. because you want TrueCrypt to use only a fully open-source implementation of AES), you can do so by selecting *Settings > Performance* and disabling the option '*Accelerate AES encryption/decryption by using the AES instructions of the processor*'. Note that when this setting is changed, the operating system needs to be restarted to ensure that all TrueCrypt components internally perform the requested change of mode. Also note that when you create a TrueCrypt Rescue Disk, the state of this option is written to the Rescue Disk and used whenever you boot from it (affecting the pre-boot and initial boot phase). To create a new TrueCrypt Rescue Disk, select *System > Create Rescue Disk*.

Note: Support for hardware acceleration was introduced in TrueCrypt 7.0.

Encryption Algorithms

TrueCrypt volumes can be encrypted using the following algorithms:

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

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

AES

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

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

Next Section >>

Serpent

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

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

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

* 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].

Next Section >>

Twofish

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

Next Section >>

AES-Twofish

Two ciphers in a cascade [15, 16] operating in XTS mode (see the section Modes of Operation). Each 128-bit block is first encrypted with Twofish (256-bit key) in XTS mode and then with AES (256-bit key) in XTS mode. Each of the cascaded ciphers uses its own key. All encryption keys are mutually independent (note that header keys are independent too, 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.

AES-Twofish-Serpent

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

Serpent-AES

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

Serpent-Twofish-AES

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

Twofish-Serpent

Two ciphers in a cascade [15, 16] operating in XTS mode (see the section Modes of Operation). Each 128-bit block is first encrypted with Serpent (256-bit key) in XTS mode and then with Twofish (256-bit key) in XTS mode. Each of the cascaded ciphers uses its own key. All encryption keys are mutually independent (note that header keys are independent too, even though they are derived from a single password – see

Hash Algorithms

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

TrueCrypt currently supports the following hash algorithms:

- **RIPEMD-160**
- **SHA-512**
- **Whirlpool**

Next Section >>

RIPEMD-160

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

Next Section >>

SHA-512

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

Next Section >>

Whirlpool

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

See also: **Encryption Algorithms**, **Technical Details**

Technical Details

Next Section >>

Notation

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

Next Section >>

Encryption Scheme

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

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

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

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

   a. PRF used by the header key derivation function (as specified in PKCS #5 v2.0; see the section Header Key Derivation, Salt, and Iteration Count), which can be one of the following:

      HMAC-SHA-512, HMAC-RIPEMD-160, HMAC-Whirlpool.

      A password entered by the user (to which one or more keyfiles may have been applied – see the section Keyfiles) and the salt read in (1) are passed to the header key derivation function, which produces a sequence of values (see the section Header Key Derivation, Salt, and Iteration Count) from which the header encryption key and secondary header key (XTS mode) are formed. (These keys are used to decrypt the volume header.)

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

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

   d. Key size(s)

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)

Modes of Operation

The mode of operation used by TrueCrypt for encrypted partitions, drives, and virtual volumes is XTS.

XTS mode is in fact XEX mode [12], which was designed by Phillip Rogaway in 2003, with a minor modification (XEX mode uses a single key for two different purposes, whereas XTS mode uses two independent keys).

In 2010, XTS mode was approved by NIST for protecting the confidentiality of data on storage devices [24]. In 2007, it was also approved by the IEEE for cryptographic protection of data on block-oriented storage devices (IEEE 1619).

Description of XTS mode:

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

Where:

$\otimes$ denotes multiplication of two polynomials over the binary field GF(2) modulo $x^{128}+x^7+x^2+x+1$

$K1$ is the encryption key (256-bit for each supported cipher; i.e, AES, Serpent, and Twofish)

$K2$ is the secondary key (256-bit for each supported cipher; i.e, AES, Serpent, and Twofish)

i is the cipher block index within a data unit; for the first cipher block within a data unit, $i = 0$

n is the data unit index within the scope of $K1$; for the first data unit, $n = 0$

a is a primitive element of Galois Field (2^{128}) that corresponds to polynomial x (i.e., 2)

Note: The remaining symbols are defined in the section Notation.

The size of each data unit is always 512 bytes (regardless of the sector size).

For further information pertaining to XTS mode, see e.g. [12] and [24].

Header Key Derivation, Salt, and Iteration Count

Header key is used to encrypt and decrypt the encrypted area of the TrueCrypt volume header (for system encryption, of the keydata area), which contains the master key and other data (see the sections Encryption Scheme and TrueCrypt Volume Format Specification). In volumes created by TrueCrypt 5.0 or later (and for system encryption), the area is encrypted in XTS mode (see the section Modes of Operation). The method that TrueCrypt uses to generate the header key and the secondary header key (XTS mode) is PBKDF2, specified in PKCS #5 v2.0; see [7].

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

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

Next Section >>

Random Number Generator

The random number generator (RNG) is used to generate the master encryption key, the secondary key (XTS mode), salt, and keyfiles. It creates a pool of random values in RAM (memory). The pool, which is 320 bytes long, is filled with data from the following sources:

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

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

Pool Mixing Function

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

Description of the pool mixing function:

1. Let *R* be the randomness pool.
2. Let *H* be the hash function selected by the user (SHA-512, RIPEMD-160, or Whirlpool).
3. *l* = byte size of the output of the hash function *H* (i.e., if *H* is RIPEMD-160, then *l* = 20; if *H* is SHA-512, *l* = 64)

Keyfiles

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

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

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

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

1. Let *P* be a TrueCrypt volume password supplied by user (may be empty)
2. Let *KP* be the keyfile pool
3. Let *kpl* be the size of the keyfile pool *KP*, in bytes (64, i.e., 512 bits);
   kpl must be a multiple of the output size of a hash function *H*
4. Let *pl* be the length of the password *P*, in bytes (in the current version: $0 \leq pl \leq 64$)
5. if $kpl > pl$, append $(kpl - pl)$ zero bytes to the password *P* (thus $pl = kpl$)
6. Fill the keyfile pool *KP* with *kpl* zero bytes.
7. For each keyfile perform the following steps:

    a. Set the position of the keyfile pool cursor to the beginning of the pool
    b. Initialize the hash function *H*
    c. Load all bytes of the keyfile one by one, and for each loaded byte perform the following steps:

        i. Hash the loaded byte using the hash function *H* without initializing the hash, to obtain an intermediate hash (state) *M.* Do not finalize the hash (the state is retained for next round).
        ii. Divide the state *M* into individual bytes.
            For example, if the hash output size is 4 bytes, $(T_0 \mid\mid T_1 \mid\mid T_2 \mid\mid T_3) = M$
        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.

TrueCrypt Volume Format Specification

Note that this specification applies to volumes created by TrueCrypt 7.0 or later. The format of file-hosted volumes is identical to the format of partition/device-hosted volumes (however, the "volume header", or key data, for a system partition/drive is stored in the last 512 bytes of the first logical drive track). TrueCrypt volumes have no "signature" or ID strings. Until decrypted, they appear to consist solely of random data.

Free space on each TrueCrypt volume is filled with random data when the volume is created.* The random data is generated as follows: Right before TrueCrypt volume formatting begins, a temporary encryption key and a temporary secondary key (XTS mode) are generated by the random number generator (see the section Random Number Generator). The encryption algorithm that the user selected is initialised with the temporary keys. The encryption algorithm is then used to encrypt plaintext blocks consisting of zeroes. The encryption algorithm operates in XTS mode (see the section Hidden Volume). The resulting ciphertext blocks are used to fill (overwrite) the free space on the volume. The temporary keys are stored in RAM and are erased after formatting finishes.

TrueCrypt Volume Format Specification:

Offset (bytes)	Size (bytes)	Encryption Status†	Description
0	64	Unencrypted§	Salt
64	4	Encrypted	ASCII string "TRUE"
68	2	Encrypted	Volume header format version (5)
70	2	Encrypted	Minimum program version required to open the volume
72	4	Encrypted	CRC-32 checksum of the (decrypted) bytes 256-511
76	16	Encrypted	Reserved (must contain zeroes)
92	8	Encrypted	Size of hidden volume (set to zero in non-hidden volumes)
100	8	Encrypted	Size of volume
108	8	Encrypted	Byte offset of the start of the master key scope
116	8	Encrypted	Size of the encrypted area within the master key scope

Compliance with Standards and Specifications

To our best knowledge, TrueCrypt complies with the following standards, specifications, and recommendations:

- ISO/IEC 10118-3:2004 [21]
- FIPS 197 [3]
- FIPS 198 [22]
- FIPS 180-2 [14]
- NIST SP 800-3E [24]
- PKCS #5 v2.0 [7]
- PKCS #11 v2.20 [23]

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

Next Section >>

Source Code

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

http://www.truecrypt.org/

TrueCrypt Volume

There are two types of TrueCrypt volumes:

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

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

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

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

Next Section >>

Creating a New TrueCrypt Volume

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

Hash Algorithm

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

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

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

Encryption Algorithm

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

Quick Format

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

Favorite Volumes

Favorite volumes are useful, for example, in any the following cases:

- You have a volume that always needs to be **mounted to a particular drive letter**.
- You have a volume that needs to be **automatically mounted when its host device gets connected to the computer** (for example, a container located on a USB flash drive or external USB hard drive).
- You have a volume that needs to be **automatically mounted when you log on** to the operating system.
- You have a volume that always needs to be **mounted as read-only** or removable medium.

To configure a TrueCrypt volume as a favorite volume, follow these steps:

1. Mount the volume (to the drive letter to which you want it to be mounted every time).
2. Right-click the mounted volume in the drive list in the main TrueCrypt window and select '*Add to Favorites*'.
3. The Favorite Volumes Organizer window should appear now. In this window, you can set various options for the volume (see below).
4. Click *OK*.

Favorite volumes can be mounted in several ways: To mount all favorite volumes, select *Favorites > Mount Favorite Volumes* or press the '*Mount Favorite Volumes*' hot key (*Settings > Hot Keys*). To mount only one of the favorite volumes, select it from the list contained in the *Favorites* menu. When you do so, you are asked for its password (and/or keyfiles) (unless it is cached) and if it is correct, the volume is mounted. If it is already mounted, an Explorer window is opened for it.

Selected or all favorite volumes can be mounted automatically whenever you log on to Windows. To set this up, follow these steps:

1. Mount the volume you want to have mounted automatically when you log on (mount it to the drive letter to which you want it to be mounted every time).
2. Right-click the mounted volume in the drive list in the main TrueCrypt window and select '*Add to Favorites*'.
3. The Favorites Organizer window should appear now. In this window, enable the option '*Mount selected volume upon logon*' and click *OK*.

System Favorite Volumes

System favorites are useful, for example, in the following cases:

- You have volumes that need to be **mounted before system and application services start and before users start logging on**.
- There are network-shared folders located on TrueCrypt volumes. If you configure these volumes as system favorites, you will ensure that the **network shares will be automatically restored** by the operating system each time it is restarted.
- You need each such volume to be mounted as **the same drive letter** each time the operating system starts.

Note that, unlike the regular (non-system) favorites, **system favorite volumes use the pre-boot authentication password** and, therefore, require your system partition/drive to be encrypted (also note it is not required to enable caching of the pre-boot authentication password).

System favorite volumes **can be configured to be available within TrueCrypt only to users with administrator privileges** (select *Settings* > '*System Favorite Volumes*' > '*Allow only administrators to view and dismount system favorite volumes in TrueCrypt*'). This option should be enabled on servers to ensure that system favorite volumes cannot be dismounted by users without administrator privileges. On non-server systems, this option can be used to prevent normal TrueCrypt volume actions (such as '*Dismount All*', auto-dismount, etc.) from affecting system favorite volumes. In addition, when TrueCrypt is run without administrator privileges (the default on Windows Vista and later), system favorite volumes will not be displayed in the drive letter list in the main TrueCrypt application window.

To configure a TrueCrypt volume as a system favorite volume, follow these steps:

1. Mount the volume (to the drive letter to which you want it to be mounted every time).
2. Right-click the mounted volume in the drive list in the main TrueCrypt window and select '*Add to System Favorites*'.
3. The System Favorites Organizer window should appear now. In this window, enable the option '*Mount system favorite volumes when Windows starts*' and click *OK*.

The order in which system favorite volumes are displayed in the System Favorites Organizer window (*Favorites* > '*Organize System Favorite Volumes*') is **the order in which the volumes are mounted**. You can use the *Move Up* and *Move Down* buttons to change the order of the volumes.

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 a 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* below 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 (except DVD drives and similar devices) one by one and tries to mount each of them as a TrueCrypt volume. Note that a 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

Program Menu

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

Volumes -> Auto-Mount All Device-Hosted Volumes

See the section *Auto-Mount Devices* in the chapter Main Program Window.

Volumes -> Dismount All Mounted Volumes

See the section *Dismount All* in the chapter Main Program Window.

Volumes -> Set Header Key Derivation Algorithm

This function allows you to re-encrypt a volume header with a header key derived using a different PRF function (for example, instead of HMAC-RIPEMD-160 you could use HMAC-SHA-512). 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 256 times with random data to prevent adversaries from using techniques such as magnetic force microscopy or magnetic force scanning tunneling microscopy [17] to recover the overwritten header (however, see also the chapter Security Requirements and Precautions).

Volumes -> Change Volume Password

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

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

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 subsection *Settings -> Preferences*, item *Cache passwords in driver memory* in the section Program Menu.

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 a favorite volume in the *Favorites* menu*,* 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.

Mount volume as removable medium

See section Volume Mounted as Removable Medium.

Use backup header embedded in volume if available

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

Mount partition using system encryption without pre-boot authentication

Supported Operating Systems

TrueCrypt currently supports the following operating systems:

- Windows 7 (32-bit and 64-bit)
- Windows Vista (32-bit and 64-bit)
- Windows XP (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit)
- Windows Server 2008 (32-bit and 64-bit)
- Windows Server 2003 (32-bit and 64-bit)
- Windows 2000 SP4

- Mac OS X 10.8 Mountain Lion (32-bit and 64-bit)
- Mac OS X 10.7 Lion (32-bit and 64-bit)
- Mac OS X 10.6 Snow Leopard (32-bit)
- Mac OS X 10.5 Leopard
- Mac OS X 10.4 Tiger

- Linux (32-bit and 64-bit versions, kernel 2.6 or compatible)

Note: The following operating systems (among others) are not supported: Windows RT, Windows 2003 IA-64, Windows 2008 IA-64, Windows XP IA-64, and the Embedded/Tablet versions of Windows.

See also: **Operating Systems Supported for System Encryption**

Portable Mode

TrueCrypt can run in so-called portable 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:

- You need administrator privileges in order to be able to run TrueCrypt in portable mode (for the reasons, see the chapter Using TrueCrypt Without Administrator Privileges).

  Note: No matter what kind of software you use, as regards personal privacy in most cases, it is *not* secure to work with sensitive data under systems where you do not have administrator privileges, as the administrator can easily capture and copy your sensitive data, including passwords and keys.

- 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 had been run in portable mode.

Note: If that is a problem, see this question in the FAQ for a possible solution.

There are two ways to run TrueCrypt in portable mode:

- After you extract files from the TrueCrypt self-extracting package, you can directly run *TrueCrypt.exe*.

  Note: To extract files from the TrueCrypt self-extracting package, run it, and then select *Extract* (instead of *Install*) on the second page of the TrueCrypt Setup wizard.

- You can use the *Traveler Disk Setup* facility to prepare a special traveler disk and launch TrueCrypt from there.

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

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

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 section Keyfiles in the chapter Technical Details). Until the correct keyfile is provided, no volume that uses the keyfile can be mounted.

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

- May improve protection against brute force attacks (significant particularly if the volume password is not very strong).
- Allows the use of security tokens and smart cards (see below).
- Allows multiple users to mount a single volume using different user passwords or PINs. Just give each user a security token or smart card containing the same TrueCrypt keyfile and let them choose their personal password or PIN that will protect their security token or smart card.
- 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) can 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. 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 *Tools > Keyfile Generator*.

Note: Keyfiles are currently not supported for system encryption.

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 subsection* 'Settings -> Preferences'*, item* 'Cache passwords in driver memory' *in the section* Program Menu).

See also the section Choosing Passwords and Keyfiles in the chapter Security Requirements and Precautions.

Keyfiles Dialog Window

Security Tokens & Smart Cards

TrueCrypt supports security (or cryptographic) tokens and smart cards that can be accessed using the PKCS #11 (2.0 or later) protocol [23]. For more information, please see the section *Security Tokens and Smart Cards* in the chapter *Keyfiles*.

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
2. 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*'.
3. Run TrueCrypt.
4. Select *Settings* -> *Language*, then select your language and click *OK*.

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

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.

See also: **Main Program Window**

Security Model

Note to security researchers: If you intend to report a security issue or publish an attack on TrueCrypt, please make sure it does not disregard the security model of TrueCrypt described below. If it does, the attack (or security issue report) will be considered invalid/bogus.

TrueCrypt is a computer software program whose primary purposes are to:

- Secure data by encrypting it before it is written to a disk.
- Decrypt encrypted data after it is read from the disk.

TrueCrypt does **not**:

- Encrypt or secure any portion of RAM (the main memory of a computer).
- Secure any data on a computer* if an attacker has administrator privileges** under an operating system installed on the computer.
- Secure any data on a computer if the computer contains any malware (e.g. a virus, Trojan horse, spyware) or any other piece of software (including TrueCrypt or an operating system component) that has been altered, created, or can be controlled, by an attacker.
- Secure any data on a computer if an attacker has physical access to the computer before or while TrueCrypt is running on it.
- Secure any data on a computer if an attacker has physical access to the computer between the time when TrueCrypt is shut down and the time when the entire contents of all volatile memory modules connected to the computer (including memory modules in peripheral devices) have been permanently and irreversibly erased/lost.
- Secure any data on a computer if an attacker can remotely intercept emanations from the computer hardware (e.g. the monitor or cables) while TrueCrypt is running on it (or otherwise remotely monitor the hardware and its use, directly or indirectly, while TrueCrypt is running on it).
- Secure any data stored in a TrueCrypt volume*** if an attacker without administrator privileges can access the contents of the mounted volume (e.g. if file/folder/volume permissions do not prevent such an attacker from accessing it).
- Preserve/verify the integrity or authenticity of encrypted or decrypted data.
- Prevent traffic analysis when encrypted data is transmitted over a network.
- Prevent an attacker from determining in which sectors of the volume the content changed (and when and how many times) if he or she can observe the volume (dismounted or mounted) before and after data is written to it, or if the storage medium/device allows the attacker to determine such information (for example, the volume resides on a device that saves metadata that can be

Security Requirements and Precautions

IMPORTANT: If you want to use TrueCrypt, you must follow the security requirements and security precautions listed in this chapter.

The sections in this chapter specify security requirements for using TrueCrypt and give information about things that adversely affect or limit the ability of TrueCrypt to secure data and to provide plausible deniability. Disclaimer: This chapter is not guaranteed to contain a list of *all* security issues and attacks that might adversely affect or limit the ability of TrueCrypt to secure data and to provide plausible deniability.

Data Leaks

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

In order to prevent data leaks, you must follow these steps (alternative steps may exist):

- If you do *not* need plausible deniability:

    - Encrypt the system partition/drive (for information on how to do so, see the chapter System Encryption) and ensure that only encrypted or read-only filesystems are mounted during each session in which you work with sensitive data.

      or,

    - If you cannot do the above, download or create a "live CD" version of your operating system (i.e. a "live" system entirely stored on and booted from a CD/DVD) that ensures that any data written to the system volume is written to a RAM disk. When you need to work with sensitive data, boot such a live CD/DVD and ensure that only encrypted and/or read-only filesystems are mounted during the session.

- If you need plausible deniability:

    - Create a hidden operating system. TrueCrypt will provide automatic data leak protection. For more information, see the section Hidden Operating System.

      or,

    - If you cannot do the above, download or create a "live CD" version of your operating system (i.e. a "live" system entirely stored on and booted from a CD/DVD) that ensures that any data written to the system volume is written to a RAM disk. When you need to work with sensitive data, boot such a live CD/DVD. If you use hidden volumes, follow the security requirements and precautions listed in the subsection Security Requirements and Precautions Pertaining to Hidden Volumes. If you do not use hidden volumes, ensure that only non-system partition-hosted TrueCrypt volumes and/or read-only filesystems are mounted during the session.

Paging File

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

Paging files, also called swap files, are used by Windows to hold parts of programs and data files that do not fit in memory. This means that sensitive data, which you believe are only stored in RAM, can actually be written *unencrypted* to a hard drive by Windows without you knowing.

Note that 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).

To prevent the issues described above, encrypt the system partition/drive (for information on how to do so, see the chapter System Encryption) and make sure that all paging files are located on one or more of the partitions within the key scope of system encryption (for example, on the partition where Windows is installed). Note that the last condition is typically met on Windows XP by default. However, Windows Vista and later versions of Windows are configured by default to create paging files on any suitable volume. Therefore, before, you start using TrueCrypt, you must follow these steps: Right-click the '*Computer*' (or '*My Computer*') icon on the desktop or in the *Start Menu*, and then select *Properties* > (on Windows Vista or later: > *Advanced System Settings* >) *Advanced* tab > section *Performance* > *Settings* > *Advanced* tab > section *Virtual memory* > *Change*. On Windows Vista or later, disable '*Automatically manage paging file size for all drives*'. Then make sure that the list of volumes available for paging file creation contains only volumes within the intended key scope of system encryption (for example, the volume where Windows is installed). To disable paging file creation on a particular volume, select it, then select '*No paging file*' and click *Set*. When done, click *OK* and restart the computer.

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

Next Section >>

Hibernation File

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

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

Note that when Windows enters Sleep mode, it may be actually configured to enter so-called Hybrid Sleep mode, which involves hibernation. Also note that the operating system may be configured to hibernate or enter the Hybrid Sleep mode when you click or select "Shut down" (for more information, please see the documentation for your operating system).

To prevent the issues described above, encrypt the system partition/drive (for information on how to do so, see the chapter System Encryption) and make sure that the hibernation file is located on one of the partitions within the key scope of system encryption (which it typically is, by default), for example, on the partition where Windows is installed. When the computer hibernates, any data will be encrypted on the fly before being written to the hibernation file.

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

Alternatively, if you cannot use system encryption, disable or prevent hibernation on your computer at least for each session during which you work with any sensitive data and during which you mount a TrueCrypt volume.

* Disclaimer: As Windows XP and Windows 2003 do not provide any API for encryption of hibernation files, TrueCrypt has to modify undocumented components of Windows XP/2003 in order to allow users to encrypt hibernation files. Therefore, TrueCrypt cannot guarantee that Windows XP/2003 hibernation files will always be

Memory Dump Files

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

Most operating systems, including Windows, can be configured to write debugging information and contents of the system memory to so-called memory dump files (also called crash 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, you must disable memory dump file generation on your computer at least for each session during which you work with any sensitive data and during which you mount a TrueCrypt volume. To do so in Windows XP or later, right-click the '*Computer*' (or '*My Computer*') icon on the desktop or in the *Start Menu*, and then select *Properties* > (on Windows Vista or later: > *Advanced System Settings* >) *Advanced* tab > section *Startup and Recovery* > *Settings* > section *Write debugging information* > select *(none)* > *OK*.

Note for users of Windows XP/2003: As Windows XP and Windows 2003 do not provide any API for encryption of memory dump files, if the system partition/drive is encrypted by TrueCrypt and your Windows XP system is configured to write memory dump files to the system drive, the TrueCrypt driver automatically prevents Windows from writing any data to memory dump files*.*

Next Section >>

Unencrypted Data in RAM

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

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

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

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

* Allegedly, for 1.5-35 seconds under normal operating temperatures (26-44 °C) and up to several hours when the memory modules are cooled (when the computer is running) to very low temperatures (e.g. -50 °C). New types of memory modules allegedly exhibit a much shorter decay time (e.g. 1.5-2.5 seconds) than older types (as of 2008).

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

Physical Security

If an attacker can physically access the computer hardware **and** you use it after the attacker has physically accessed it, then TrueCrypt may become unable to secure data on the computer.* This is because the attacker may modify the hardware or attach a malicious hardware component to it (such as a hardware keystroke logger) that will capture the password or encryption key (e.g. when you mount a TrueCrypt volume) or otherwise compromise the security of the computer. Therefore, you must not use TrueCrypt on a computer that an attacker has physically accessed. Furthermore, you must ensure that TrueCrypt (including its device driver) is not running when the attacker physically accesses the computer. Additional information pertaining to hardware attacks where the attacker has direct physical access is contained in the section Unencrypted Data in RAM.

Furthermore, even if the attacker cannot physically access the computer hardware *directly*, he or she may be able to breach the physical security of the computer by remotely intercepting and analyzing emanations from the computer hardware (including the monitor and cables). For example, intercepted emanations from the cable connecting the keyboard with the computer can reveal passwords you type. It is beyond the scope of this document to list all of the kinds of such attacks (sometimes called TEMPEST attacks) and all known ways to prevent them (such as shielding or radio jamming). It is your responsibility to prevent such attacks. If you do not, TrueCrypt may become unable to secure data on the computer.

* In this section (*Physical Security*), the phrase "data on the computer" means data on internal and external storage devices/media (including removable devices and network drives) connected to the computer.

Next Section >>

Malware

The term 'malware' refers collectively to all types of malicious software, such as computer viruses, Trojan horses, spyware, or generally any piece of software (including TrueCrypt or an operating system component) that has been altered, prepared, or can be controlled, by an attacker. Some kinds of malware are designed e.g. to log keystrokes, including typed passwords (such captured passwords are then either sent to the attacker over the Internet or saved to an unencrypted local drive from which the attacker might be able to read it later, when he or she gains physical access to the computer). If you use TrueCrypt on a computer infected with any kind of malware, TrueCrypt may become unable to secure data on the computer.* Therefore, you must *not* use TrueCrypt on such a computer.

It is important to note that TrueCrypt is encryption software, *not* anti-malware software. It is your responsibility to prevent malware from running on the computer. If you do not, TrueCrypt may become unable to secure data on the computer.

There are many rules that you should follow to help prevent malware from running on your computer. Among the most important rules are the following: Keep your operating system, Internet browser, and other critical software, up-to-date. In Windows XP or later, turn on DEP for all programs.** Do not open suspicious email attachments, especially executable files, even if they appear to have been sent by your relatives or friends (their computers may be infected with malware sending malicious emails from their computers/accounts without their knowledge). Do not follow suspicious links contained in emails or on websites (even if the email/website appears to be harmless or trustworthy). Do not visit any suspicious websites. Do not download or install any suspicious software. Consider using good, trustworthy, anti-malware software.

* In this section (*Malware*), the phrase "data on the computer" means data on internal and external storage devices/media (including removable devices and network drives) connected to the computer.
** DEP stands for Data Execution Prevention. For more information about DEP, please visit http://support.microsoft.com/kb/875352, http://technet.microsoft.com/en-us/library/cc700810.aspx, and http://windows.microsoft.com/en-US/windows7/What-is-Data-Execution-Prevention.

Next Section >>

Multi-User Environment

Keep in mind, that the content of a mounted TrueCrypt volume is visible (accessible) to all logged on users. NTFS file/folder permissions can be set to prevent this, unless the volume is mounted as removable medium under a desktop edition of Windows Vista or later (sectors of a volume mounted as removable medium may be accessible at the volume level to users without administrator privileges, regardless of whether it is accessible to them at the file-system level).

Moreover, on Windows, the password cache is shared by all logged on users.

Also note that switching users in Windows XP or later (*Fast User Switching* functionality) does *not* dismount a successfully mounted TrueCrypt volume (unlike system restart, which dismounts all mounted TrueCrypt volumes).

On Windows 2000, the container file permissions are ignored when a file-hosted TrueCrypt volume is to be mounted. On all supported versions of Windows, users without administrator privileges can mount any partition/device-hosted TrueCrypt volume (provided that they supply the correct password and/or keyfiles). A user without administrator privileges can dismount only volumes that he or she mounted. However, this does not apply to system favorite volumes unless you enable the option (disabled by default) *Settings* > '*System Favorite Volumes*' > '*Allow only administrators to view and dismount system favorite volumes in TrueCrypt*'.

Next Section >>

Authenticity and Integrity

TrueCrypt uses encryption to preserve the *confidentiality* of data it encrypts. TrueCrypt neither preserves nor verifies the *integrity* or *authenticity* of data it encrypts or decrypts. Hence, if you allow an adversary to modify data encrypted by TrueCrypt, he can set the value of any 16-byte block of the data to a random value or to a previous value, which he was able to obtain in the past. Note that the adversary cannot choose the value that you will obtain when TrueCrypt decrypts the modified block — the value will be random — unless the attacker restores an older version of the encrypted block, which he was able to obtain in the past. It is your responsibility to verify the integrity and authenticity of data encrypted or decrypted by TrueCrypt (for example, by using appropriate third-party software).

See also: Physical Security, Security Model

Next Section >>

Choosing Passwords and Keyfiles

It is very important that you choose a good password. You must avoid choosing one that contains only a single word that can be found in a dictionary (or a combination of such words). It must not contain any names, dates of birth, account numbers, or any other items that could be easy to guess. A good password is a random combination of upper and lower case letters, numbers, and special characters, such as @ ^ = $ * + etc. We strongly recommend choosing a password consisting of more than 20 characters (the longer, the better). Short passwords are easy to crack using brute-force techniques.

To make brute-force attacks on a keyfile infeasible, the size of the keyfile must be at least 30 bytes. If a volume uses multiple keyfiles, then at least one of the keyfiles must 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 must 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 *Tools -> Keyfile Generator*).

When creating a volume, encrypting a system partition/drive, or changing passwords/keyfiles, you must not allow any third party to choose or modify the password/keyfile(s) before/while the volume is created or the password/keyfiles(s) changed. For example, you must not use any password generators (whether website applications or locally run programs) where you are not sure that they are high-quality and uncontrolled by an attacker, and keyfiles must not be files that you download from the internet or that are accessible to other users of the computer (whether they are administrators or not).

Next Section >>

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 (not to be confused with the password) 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 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.

The following sections of this chapter contain additional information pertaining to possible security issues connected with changing passwords and/or keyfiles:

- Wear-Leveling
- Journaling File Systems
- Defragmenting
- Reallocated Sectors

Next Section >>

Trim Operation

Some storage devices (e.g., some solid-state drives, including USB flash drives) use so-called 'trim' operation to mark drive sectors as free e.g. when a file is deleted. Consequently, such sectors may contain unencrypted zeroes or other undefined data (unencrypted) even if they are located within a part of the drive that is encrypted by TrueCrypt. TrueCrypt does not block the trim operation on partitions that are within the key scope of system encryption (unless a hidden operating system is running) and under Linux on all volumes that use the Linux native kernel cryptographic services. In those cases, the adversary will be able to tell which sectors contain free space (and may be able to use this information for further analysis and attacks) and plausible deniability may be negatively affected. If you want to avoid those issues, do not use system encryption on drives that use the trim operation and, under Linux, either configure TrueCrypt not to use the Linux native kernel cryptographic services or make sure TrueCrypt volumes are not located on drives that use the trim operation.

To find out whether a device uses the trim operation, please refer to documentation supplied with the device or contact the vendor/manufacturer.

Next Section >>

Wear-Leveling

Some storage devices (e.g., some solid-state drives, including USB flash drives) and some file systems utilize so-called wear-leveling mechanisms to extend the lifetime of the storage device or medium. These mechanisms ensure that even if an application repeatedly writes data to the same logical sector, the data is distributed evenly across the medium (logical sectors are remapped to different physical sectors). Therefore, multiple "versions" of a single sector may be available to an attacker. This may have various security implications. For instance, when you change a volume password/keyfile(s), the volume header is, under normal conditions, overwritten with a re-encrypted version of the header. However, when the volume resides on a device that utilizes a wear-leveling mechanism, TrueCrypt cannot ensure that the older header is really overwritten. If an adversary found the old volume header (which was to be overwritten) on the device, he could use it to mount the volume using an old compromised password (and/or using compromised keyfiles that were necessary to mount the volume before the volume header was re-encrypted). Due to security reasons, we recommend that TrueCrypt volumes are not created/stored on devices (or in file systems) that utilize a wear-leveling mechanism (and that TrueCrypt is not used to encrypt any portions of such devices or filesystems).

If you decide not to follow this recommendation and you intend to use in-place encryption on a drive that utilizes wear-leveling mechanisms, make sure the partition/drive does not contain any sensitive data before you fully encrypt it (TrueCrypt cannot reliably perform secure in-place encryption of existing data on such a drive; however, after the partition/drive has been fully encrypted, any new data that will be saved to it will be reliably encrypted on the fly). That includes the following precautions: Before you run TrueCrypt to set up pre-boot authentication, disable the paging files and restart the operating system (you can enable the paging files after the system partition/drive has been fully encrypted). Hibernation must be prevented during the period between the moment when you start TrueCrypt to set up pre-boot authentication and the moment when the system partition/drive has been fully encrypted. However, note that even if you follow those steps, it is *not* guaranteed that you will prevent data leaks and that sensitive data on the device will be securely encrypted. For more information, see the sections Data Leaks, Paging File, Hibernation File, and Memory Dump Files.

If you need plausible deniability, you must not use TrueCrypt to encrypt any part of (or create encrypted containers on) a device (or file system) that utilizes 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.

Next Section >>

Reallocated Sectors

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

Next Section >>

Defragmenting

When you (or the operating system) defragment the file system in which a file-hosted TrueCrypt container is stored, 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 and other possible security issues (such as those mentioned in the section Volume Clones), 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.

Next Section >>

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 that were necessary to mount the volume before the volume header was re-encrypted). Some journaling file systems also internally record file access times and other potentially sensitive information. If you need plausible deniability, you must not store file-hosted TrueCrypt containers in journaling file systems. To prevent possible security issues related to journaling file systems, do one the following:

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

Next Section >>

Volume Clones

Never create a new TrueCrypt volume by cloning an existing TrueCrypt volume. Always use the TrueCrypt Volume Creation Wizard to create a new TrueCrypt volume. If you clone a volume and then start using both this volume and its clone in a way that both eventually contain different data, then you might aid cryptanalysis (both volumes will share a single key set). Also note that plausible deniability is impossible in such cases. See also the chapter How to Back Up Securely.

Next Section >>

Additional Security Requirements and Precautions

In addition to the requirements and precautions described in this chapter (Security Requirements and Precautions), you must follow and keep in mind the security requirements, precautions, and limitations listed in the following chapters and sections:

- **How to Back Up Securely**
- **Limitations**
- **Security Model**
- **Security Requirements and Precautions Pertaining to Hidden Volumes**
- **Plausible Deniability**

See also: **Digital Signatures**

Command Line Usage

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

`/help` or `/?`

Display command line help.

`/volume` or `/v`

Path to a TrueCrypt volume to mount (do not use when dismounting). For a file-hosted volume, the path must include the filename. To mount a partition/device-hosted volume, use, for example, `/v \Device\Harddisk1\Partition3` (to determine the path to a partition/device, run TrueCrypt and click *Select Device*). You can also mount a partition or dynamic volume using its volume name (for example, `/v \\?\Volume{5cceb196-48bf-46ab-ad00-70965512253a}\`). To determine the volume name use e.g. `mountvol.exe`. Also 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 designated as "mount upon logon". Note that `/auto` is implicit if `/quit` and `/volume` are specified. If you need to prevent the application window from appearing, use `/quit`.

How to Back Up Securely

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

Non-System Volumes

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

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

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

2. Mount the newly created *backup* volume.

3. Mount the *main* volume.

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

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

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

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

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

Miscellaneous

Using TrueCrypt Without Administrator Privileges

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

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

Warning: No matter what kind of software you use, as regards personal privacy in most cases, it is *not* secure to work with sensitive data under systems where you do not have administrator privileges, as the administrator can easily capture and copy your sensitive data, including passwords and keys.

Sharing over Network

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

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

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

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

   Remarks: Note that, when you restart the system, the network share will be automatically restored only if the volume is a system favorite volume or an encrypted system partition/drive (for information on how to configure a volume as a system favorite volume, see the chapter System Favorite Volumes).

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

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

   Disadvantages: The shared volume may be only file-hosted (not partition/device-hosted). The volume must be mounted in read-only mode under each of the systems (see the section Mount Options for information on how to mount a volume in read-only mode). Note that this requirement applies to unencrypted volumes too. 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).

See also: **Using TrueCrypt without Administrator Privileges**

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 logoff, inadvertent host device removal, time-out, etc.)
3) Auto-mount of favorite volumes
4) Notifications (e.g., when damage to hidden volume is prevented)
5) 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.

See also: **Main Program Window**

Volume Mounted as Removable Medium

This section applies to TrueCrypt volumes mounted when one of the following options is enabled (as applicable):

- *Tools* > *Preferences* > *Mount volumes as removable media*
- *Mount Options* > *Mount volume as removable medium*
- *Favorites* > *Organize Favorite Volumes* > *Mount selected volume as removable medium*
- *Favorites* > *Organize System Favorite Volumes* > *Mount selected volume as removable medium*

TrueCrypt Volumes that are mounted as removable media have the following advantages and disadvantages:

- Windows is prevented from automatically creating the '*Recycled*' and/or the '*System Volume Information*' folders on TrueCrypt volumes (in Windows, these folders are used by the Recycle Bin and System Restore features).
- Windows may use caching methods and write delays that are normally used for removable media (for example, USB flash drives). This might slightly decrease the performance but at the same increase the likelihood that it will be possible to dismount the volume quickly without having to force the dismount.
- The operating system may tend to keep the number of handles it opens to such a volume to a minimum. Hence, volumes mounted as removable media might require fewer forced dismounts than other volumes.
- Under Windows Vista and earlier, the '*Computer*' (or '*My Computer*') list does not show the amount of free space on volumes mounted as removable (note that this is a Windows limitation, not a bug in TrueCrypt).
- Under desktop editions of Windows Vista or later, sectors of a volume mounted as removable medium may be accessible to *all* users (including users without administrator privileges; see section Multi-User Environment).
- The operating system may use a different file indexing policy (used e.g. for instant search) for volumes mounted as removable media.

TrueCrypt System Files & Application Data

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

TrueCrypt Driver

`%windir%\SYSTEM32\DRIVERS\truecrypt.sys`

Note: This file is not present when TrueCrypt is run in portable mode.

TrueCrypt Settings, Application Data, and Other System Files

WARNING: Note that TrueCrypt does *not* encrypt any of the files listed in this section (unless it encrypts the system partition/drive).

The following files are saved in the folder `%APPDATA%\TrueCrypt\`. In portable 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):

`Configuration.xml` (the main configuration file).

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

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

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

`History.xml` (the list of last twenty files/devices attempted to be mounted as TrueCrypt volumes or attempted to be used as hosts for TrueCrypt volumes; this feature can be disabled – for more information, see the subsection *Never Save History* in the chapter Main Program Window).

How to Remove Encryption

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

1. Mount your TrueCrypt volume.
2. Move all files from the TrueCrypt volume to any location outside the TrueCrypt volume (note that the files will be decrypted on the fly).
3. Dismount the TrueCrypt volume.
4. **If the TrueCrypt volume is file-hosted**, delete it (the container) just like you delete any other file.

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

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

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

   a. Right-click the '*Computer*' (or '*My Computer*') icon on your desktop, or in the Start Menu, and select *Manage*. The '*Computer Management*' window should appear.
   b. In the *Computer Management* window, from the list on the left, select '*Disk Management*' (within the *Storage* sub-tree).
   c. The '*Initialize Disk*' window should appear. Use it to initialize the disk.
   d. In the '*Computer Management*' window, right-click the area representing the storage space of the encrypted device and select '*New Partition*' or '*New Simple Volume*'.
   e. WARNING: Before you continue, make sure you have selected the correct device, as all files stored on it will be lost. The '*New Partition Wizard*' or '*New Simple Volume Wizard*' window should appear now; follow its instructions to create a new partition on the device. After the

Uninstalling TrueCrypt

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

To uninstall TrueCrypt on Windows Vista or later, select *Start* menu > *Computer* > *Uninstall or change a program* > *TrueCrypt* > *Uninstall*.

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 portable mode.

See also: **How to Remove Encryption**

Digital Signatures

Why Verify Digital Signatures

It might happen that a TrueCrypt installation package you download from our server was created or modified by an attacker. For example, the attacker could exploit a vulnerability in the server software we use and alter the installation packages stored on the server, or he/she could alter any of the files en route to you.

Therefore, you should always verify the integrity and authenticity of each TrueCrypt distribution package you download or otherwise obtain from any source. In other words, you should always make sure that the file was created by us and it was not altered by an attacker. One way to do so is to verify so-called digital signature(s) of the file.

Types of Digital Signatures We Use

We currently use two types of digital signatures:

- **PGP** signatures (available for all binary and source code packages for all supported systems).
- **X.509** signatures (available for binary packages for Windows).

Advantages of X.509 Signatures

X.509 signatures have the following advantages, in comparison to PGP signatures:

- It is much easier to verify that the key that signed the file is really ours, not attacker's (provided that you trust the certification authority).
- You do not have to download or install any extra software to verify an X.509 signature (see below).
- You do not have to download and import our public key (it is embedded in the signed file).
- You do not have to download any separate signature file (the signature is embedded in the signed file).

Advantages of PGP Signatures

Troubleshooting

This section presents possible solutions to common problems that you may run into when using TrueCrypt.

Note: If your problem is not listed here, it might be listed in one of the following sections:

- Incompatibilities
- Known Issues & Limitations
- Frequently Asked Questions

Make sure you use the latest stable version of TrueCrypt. If the problem is caused by a bug in an old version of TrueCrypt, it may have already been fixed. Note: Select ***Help*** > ***About*** to find out which version you use.

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

Incompatibilities

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

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

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

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

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

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

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

Please note that this not a bug in TrueCrypt (the issue is caused by inappropriate design of the third-party activation software).

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

See also: **Known Issues & Limitations**, **Troubleshooting**

Known Issues & Limitations

Last Updated February 7, 2012

Known Issues

(There are currently no confirmed issues.)

Limitations

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

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

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

- To work around a Windows XP issue, the TrueCrypt boot loader is always automatically configured for the version of the operating system under which it is installed. When the version of the system changes (for example, the TrueCrypt boot loader is installed when Windows Vista is running but it is later used to boot Windows XP) you may encounter various known and unknown issues (for example, on some notebooks, Windows XP may fail to display the log-on screen). Note that this affects multi-boot configurations, TrueCrypt Rescue Disks, and decoy/hidden operating systems (therefore, if the hidden system is e.g. Windows XP, the decoy system should be Windows XP too).

- The ability to mount a partition that is within the key scope of system encryption without pre-boot authentication (for example, a partition located on the encrypted system drive of another operating system that is not running), which can be done e.g. by selecting *System > Mount Without Pre-Boot Authentication,* is limited to primary partitions (extended/logical partitions cannot be mounted this way).

- 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

License

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

http://www.truecrypt.org/legal/license

See also: **Acknowledgements**

Future Development

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

Acknowledgements

We would like to thank the following people:

Paul Le Roux for making his E4M source code available. TrueCrypt 1.0 was derived from E4M and some parts of the E4M source code are still incorporated in the latest version of the TrueCrypt source code.

Dr. Brian Gladman, who wrote the excellent AES, Twofish, and SHA-512 routines.

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

Wei Dai, who wrote the Serpent and RIPEMD-160 routines.

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

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

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

Thank you very much.

7.1a

February 7, 2012

Improvements and bug fixes:

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

7.1

September 1, 2011

New features:

- Full compatibility with 64-bit and 32-bit Mac OS X 10.7 Lion

Improvements and bug fixes:

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

7.0a

September 6, 2010

Improvements:

- Workaround for a bug in some custom (non-Microsoft) drivers for storage device controllers that caused a system crash when initiating hibernation on TrueCrypt-encrypted operating systems. (*Windows 7/Vista/2008/2008R2*)
- Other minor improvements (*Windows, Mac OS X, and Linux*)

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://csrc.nist.gov/groups/STM/cmvp/documents/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, Journal of Research of the National Institute of Standards and Technology, Vol. 106, No. 3, May-June 2001, available at http://nvl.nist.gov/pub/nistpubs/jres/106/3/j63nec.pdf.

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

[6] Bruce Schneier, *Beyond Fear: Thinking Sensibly About Security in an Uncertain World*, Springer, 2003

[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.

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

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