

# TRUECRYPT

FREE OPEN-SOURCE ON-THE-FLY ENCRYPTION FOR WINDOWS

## USER'S GUIDE

### Version Information

TrueCrypt User's Guide, version 1.0. Released February 2, 2004.

### Trademark Information

IDEA is a trademark of Ascom Tech AG. All other registered and unregistered trademarks in this document are the sole property of their respective owners.

### Licensing and Patent Information

The IDEA cryptographic cipher described in U.S. patent number 5,214,703 requires license from Ascom Tech AG for commercial use.

### Limitations

TrueCrypt Team does not warrant that the information meets your requirements or that the information is free of errors. The information may include technical inaccuracies or typographical errors.

### Disclaimer

Before using and installing this product (TrueCrypt), you must agree to the following terms and conditions:

- ! You understand that this product utilizes strong cryptography, and that should this technology be regulated or illegal in your country, the authors of the product will not be responsible for any consequences that your importing and/or using it in such a country might have.
- ! You understand that the authors of the product cannot be held responsible for loss of your data or any other damage, direct or indirect, caused by using or installing the product.
- ! You understand that this product contains no "backdoor", that would allow partial or complete recovery of your data without knowing the correct password or encryption key.
- ! You understand that the authors of the program cannot help you to recover your data should you forget your password.

You must also agree to the TrueCrypt License displayed in the TrueCrypt Setup window.

# CONTENTS

<b>INTRODUCTION.....</b>	<b>4</b>
<b>TRUECRYPT VOLUME.....</b>	<b>4</b>
CREATING A NEW TRUECRYPT VOLUME .....	4
Quick Format .....	4
Cluster Size .....	5
Additional Notes on Volume Creation.....	5
PLAUSIBLE DENIABILITY .....	5
<b>MAIN PROGRAM WINDOW .....</b>	<b>6</b>
Select File.....	6
Select Device.....	6
Mount .....	6
Auto-Mount Partitions .....	6
Dismount.....	7
Dismount All.....	7
Wipe Cache.....	7
Change Password .....	7
Never Save History .....	7
Exit.....	7
PASSWORD ENTRY .....	7
Cache Password in Driver Memory .....	7
<b>PROGRAM MENU .....</b>	<b>8</b>
File -> Exit .....	8
Tools -> Clear Volume History .....	8
Tools -> Preferences .....	8
<b>SUPPORTED OPERATING SYSTEMS.....</b>	<b>9</b>
<b>UNINSTALLING TRUECRYPT .....</b>	<b>9</b>
<b>COMMAND LINE USAGE.....</b>	<b>9</b>
Syntax .....	9
Examples.....	10
<b>ENCRYPTION ALGORITHMS.....</b>	<b>10</b>
<b>TECHNICAL DETAILS.....</b>	<b>11</b>
ENCRYPTION SCHEME .....	11
TRUECRYPT VOLUME FORMAT SPECIFICATION .....	12
HEADER KEY DERIVATION FUNCTION .....	12
SECTOR SCRAMBLING .....	13
RANDOM NUMBER GENERATOR .....	13
COMPLIANCE WITH STANDARDS .....	14
TRUECRYPT SYSTEM FILES.....	14
Windows 2000/XP .....	14

Windows 98/ME .....	15
<b>KNOWN BUGS AND DEVICE DRIVER LIMITATIONS .....</b>	<b>15</b>
Windows 2000/XP .....	15
Windows 98/ME .....	15
<b>FREQUENTLY ASKED QUESTIONS.....</b>	<b>16</b>
<b>VERSION HISTORY .....</b>	<b>16</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>19</b>
<b>CONTACT.....</b>	<b>19</b>

## PREFACE

This document assumes that the reader is generally familiar with using computer hardware and software. Describing a feature that is usually easily understood has been avoided wherever possible. For more information, please visit [www.truecrypt.org](http://www.truecrypt.org).

## Introduction

TrueCrypt is a software system for establishing and maintaining an on-the-fly-encrypted volume (data storage device). On-the-fly encryption means that data are automatically encrypted or decrypted right before they are loaded or saved, without any user intervention. *No* data stored on an encrypted volume can be read without using the correct password or encryption key. Until decrypted, encrypted volume appears to be nothing more than a series of random numbers. Entire file system is encrypted (i.e. file names, folder names, contents of every file, and free space). *No* unencrypted data are ever stored on any storage device (they are only temporarily kept in RAM during the encryption/decryption process).

## TrueCrypt Volume

There are two basic types of TrueCrypt volumes:

- ! Container
- ! Partition/device

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

A TrueCrypt *partition* is a hard disk partition encrypted using TrueCrypt. You can also encrypt floppy disks, ZIP disks, USB hard disks and other types of storage devices that allow read/write access.

## Creating a New TrueCrypt Volume

To create a new TrueCrypt volume, click on 'Create Volume' in the main program window. TrueCrypt Volume Creation Wizard should appear. The Wizard provides help and information necessary to successfully create a new TrueCrypt volume. However, several options deserve further explanation:

### Quick Format

If unchecked, each sector of the new volume will be formatted. Basically 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.

## Cluster Size

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

*Important:* Windows 98/ME do not support cluster sizes larger than 32K. You will not be able to access a 64K-cluster volume using these operating systems.

TrueCrypt volumes can be reformatted at any later time as FAT12, FAT16, FAT32, or NTFS. They behave as standard disk devices so you can right-click the device icon and select 'Format'.

## Additional Notes on Volume Creation

After you click the 'Finish' button in the wizard, there will be a short delay while your system is being polled for additional random data, which will be used in generating the new volume. TrueCrypt "hooks" the keyboard and mouse from the moment that the volume creation wizard is started. All key presses, mouse movements, and various system calls are all contributing to the random data collection. The random data are used to create volume encryption key.

*Important:* When encrypting an entire hard drive partition or an entire device (floppy disk, ZIP disk etc.), all data stored on the device/partition will be lost.

## Plausible Deniability

It is impossible to identify a TrueCrypt container or partition. Until decrypted, a TrueCrypt volume appears to consist of nothing more than random data (it does not contain any "signature"). Therefore, it is impossible to prove that a file, a partition or a device is a TrueCrypt volume and/or that it has been encrypted.

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

When formatting a hard disk partition as a TrueCrypt volume, the partition table (including the partition type) is *never* modified. If you intend to use a TrueCrypt partition and you need plausible deniability, follow these steps (Windows XP):

- 1) Right-click *My Computer* icon on your desktop and select *Manage*
- 2) In the list (on the left) click *Disk Management* (the *Storage* sub-tree)
- 3) If the partition that you want to format as a TrueCrypt has already been created, right-click it and select *Delete Partition...* If the partition has not yet been created, continue with step 4)

- 4) Right-click the free space (should be labeled as *Unallocated*) and select *New Partition...*
  - 5) *New Partition Wizard* should appear now. Follow its instructions. On the Wizard page called '*Assign Drive Letter or Path*' select '*Do not assign a drive letter or drive path*'. Click *Next*.
  - 6) Select *Do not format this partition* and click *Next*.
  - 7) Click *Finish*.
  - 8) The partition now appears to be "reserved" for future use (and future reformatting). As it is unformatted, it can contain any random data, which might, for example, have resided on the hard drive since the last time you repartitioned the hard disk. Therefore, there is no difference between such an *unformatted* partition and a TrueCrypt volume. Now you can format the partition as a TrueCrypt.
- Note: if, instead of an *unformatted* partition, you format an NTFS/FAT16/FAT32 partition as a TrueCrypt, the partition will then appear to be a corrupted NTFS/FAT16/FAT32 partition.

## Main Program Window

### Select File

Allows you to select a file-hosted TrueCrypt volume. After you select it, you can mount it by clicking 'Mount' (see below). It is also possible to select a volume by dragging its icon to the 'TrueCrypt.exe' icon (TrueCrypt will be automatically launched then).

### Select Device

Allows you to select a TrueCrypt partition or a storage device (such as floppy disk or ZIP disk). After it is selected, it can be mounted by clicking 'Mount' (see below). Note: There is a more comfortable way of mounting TrueCrypt partitions – see 'Auto-Mount Partitions' for more information.

### Mount

To mount a TrueCrypt volume, select a free drive letter from the list in the main window. Then select a file or device that hosts the TrueCrypt volume and click 'Mount'. TrueCrypt now tries to mount the volume using cached passwords (if there are any) and if none of them works, it asks you to enter a password. If you enter the correct password, the volume will be mounted.

Note that switching users on Windows XP/2000 does *not* dismount a successfully mounted TrueCrypt volume. Also note that when you exit the TrueCrypt application, the TrueCrypt driver still continues working and no TrueCrypt volumes are dismounted.

### Auto-Mount Partitions

This function allows you to mount TrueCrypt partitions without having to select them manually (by clicking 'Select Device'). TrueCrypt goes through all available partitions (on all hard drives) one by one and tries to mount each of them as a TrueCrypt volume. Note that TrueCrypt partition 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 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. Drive letters will be assigned starting from the one that is selected in the drive list in the main window. If the password you enter is not correct, mounting is tried using cached passwords (if there are any). If you enter empty password, only the cached passwords will be used when attempting to mount partitions.

## **Dismount**

To dismount a TrueCrypt volume basically means to make any access to the data it contains impossible. To do so, select a TrueCrypt volume and click on Dismount.

## **Dismount All**

Dismounts all currently mounted TrueCrypt volumes.

## **Wipe Cache**

Clears any passwords cached in driver memory. When there are no passwords in the cache, this button is disabled. Up to last four successfully mounted TrueCrypt volume passwords can be cached. This allows mounting volumes without having to type their passwords repeatedly. Passwords are *never* saved on any disk – they are only temporarily stored in RAM. Driver memory is never swapped to disk. Password caching can be enabled/disabled in the Preferences (Tools menu).

## **Change Password**

Allows changing the password of the currently selected TrueCrypt volume. The main encryption key remains unchanged. Therefore, reformatting is not necessary and is *not* performed (i.e. *no* data will be lost after changing the password and the password change will only take a few seconds).

## **Never Save History**

If checked, the file names and paths of the last eight mounted volumes will not be saved in the history.

## **Exit**

Terminates the TrueCrypt application. The driver continues working and no TrueCrypt volumes are dismounted.

## **Password Entry**

### **Cache Password in Driver Memory**

When checked, the volume password you enter will be cached in driver memory (if the password is correct). Then, later, volumes can be mounted using the cached password without having to type it

again. Up to four passwords can be cached. The passwords are *never* saved on any disk. They are only temporarily stored in RAM. Driver memory is never swapped to disk.

## Program Menu

*Note: Only the menu items that are not self-explanatory are described in this documentation.*

### **File -> Exit**

Terminates the TrueCrypt application. The driver continues working and no TrueCrypt volumes are dismounted.

### **Tools -> Clear Volume History**

Clears the list containing file names and paths of the last eight successfully mounted TrueCrypt volumes.

### **Tools -> Preferences**

#### ***Wipe cached passwords on exit***

If enabled, passwords cached in driver memory will be cleared when exiting TrueCrypt.

#### ***Cache passwords in driver memory***

When checked, up to last four successfully mounted TrueCrypt volume passwords will be cached in driver memory. Then, later, volumes can be mounted using a cached password without having to type it again. The passwords are *never* saved on any disk. They are only temporarily stored in RAM. Driver memory is never swapped to disk.

#### ***Open Explorer window for successfully mounted volume***

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

#### ***Close all Explorer windows of volume being dismounted***

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



## Supported Operating Systems

TrueCrypt runs on the following operating systems:

- ! Windows 2003
- ! Windows XP
- ! Windows 2000
- ! Windows ME
- ! Windows 98

## Uninstalling TrueCrypt

To uninstall TrueCrypt, open the Windows control panel and select 'Add/Remove Programs', locate TrueCrypt and click the 'Add/Remove' button.

Normally, all TrueCrypt files, including the device driver, should be removed, and most of the changes made to the registry should be undone. The uninstall will *never* remove any TrueCrypt volume you may have created.

## Command Line Usage

/help or /?	displays command line help
/volume or /v	file and path name of the volume to mount
/letter or /l	driver letter to mounted the volume as
/explore or /e	opens an Explorer window after a volume has been mounted
/beep or /b	beeps after a volume has been mounted
/auto or /a	automatically mounts the volume
/dismountall or /d	dismounts all currently mounted TrueCrypt volumes
/cache or /c	enables (Y) or disables (N) the password cache
/history or /h	enables (Y) or disables (N) the history
/wipecache or /wc	wipes any passwords cached in the driver memory
/password or /p	the volume password. If used with parameter of '?', a password prompt appears. This increases the level of your privacy in a multi-user environment by not displaying the main TrueCrypt window. Instead, only a small password dialog box appears.

### Syntax

```
volmount [[/v] volume] [/sd] [/l letter] [/e] [/b] [/p [?] | [password]]  
[/h] etc.
```

The order of the parameters is not important. Whitespaces between parameters and parameter values do not matter.

## Examples

Mounting a volume called 'myvolume.tcl' using the password 'MyPassword', drive letter X, TrueCrypt will open an explorer window and beep, mounting will be automatic:

```
volmount /v myfile.tc /lx /a /p MyPassword /e /b
```

Mounting a volume called 'myvolume.tc' using the password prompt:

```
volmount /v myfile.tc /lx /a /p ? /e /b
```

Note that turning the cache off will not clear the password cache. Use both /c and /wc to turn the cache off, and clear it.

## Encryption Algorithms

TrueCrypt volumes can be encrypted using one of the following algorithms:

Algorithm	Author(s)	Key Size (bits)	Block Size (bits)
Blowfish	B. Schneier	448	64
CAST	C. Adams, S. Tavares	128	64
IDEA	X. Lai, J. Massey	128	64
Triple-DES	IBM, NSA	168	64

Each of the encryption algorithms is used in CBC mode (Triple-DES in inner-CBC). A random value, unique to each sector and volume, is used as the IV (for more information, see *Sector Scrambling*).

# Technical Details

## Encryption Scheme

When mounting a TrueCrypt volume (assume there are no cached passwords), the following steps are performed:

1. The first 512 bytes of the volume are read into RAM, out of which the first 64 bytes are the salt.
2. A password entered by the user and the salt read in (1) are passed to the key derivation function (see section *Header Key Derivation Function*). This function produces a sequence of values from which the header encryption key and IV (used to decrypt the volume header) are derived.
3. An encryption algorithm is chosen and initialized with the key and IV obtained in (2). (Note that it is impossible to directly determine the cipher algorithm that has been used to encrypt the volume.)
4. The data read in (1), except the first 64 bytes, are decrypted with the chosen encryption algorithm. Note that now it is still unsure whether the chosen encryption algorithm is correct or not.
5. If the first 4 bytes of the data decrypted in (4) do not contain the text string "TRUE", then the encryption algorithm is assumed to be incorrect. If there is an algorithm that decrypting has not yet been attempted with, the process continues from (3) choosing such algorithm. If there are no algorithms remaining, the password is assumed to be incorrect, and mounting is terminated.

If the first 4 bytes of the data decrypted in (4) contain the text string "TRUE", then a CRC-32 checksum of the last 256 bytes of the data read in (1) is calculated. If this value matches the value stored at the 8<sup>th</sup> byte (see section *TrueCrypt Volume Format Specification*) of the data decrypted in (4), then the password and chosen encryption algorithm are considered correct and the process continues with (6). If the values do not match, then the encryption algorithm is assumed to be incorrect. If there is an algorithm that decrypting has not yet been attempted with, the process continues from (3) choosing such algorithm. If there are no algorithms remaining, the password is assumed to be incorrect and mounting is terminated.

6. Now we know that we have the correct password and the correct encryption algorithm. The minimum program version required to open the volume, stored in data decrypted in (4), is checked. If it is not equal or less than the version of the program that we are using to mount the volume, mounting is terminated.
7. The encryption routine is reinitialized with the master key retrieved from the data decrypted in (4). This key can be used to decrypt any sector of the volume, except the first one (the volume header, which has been encrypted using the header key).  
Note: The master key was generated during the volume creation and cannot be changed later. Volume password change is accomplished by re-encrypting the volume header using a new header key (derived from a new password).
8. Now the volume is mounted (registered with the operating system).

For more information pertaining the encryption scheme, see sections *Sector Scrambling* and *Header Key Derivation Function*.

## TrueCrypt Volume Format Specification

TrueCrypt volume has no “signature”. Until decrypted, it appears to consist of nothing more than random data. Therefore, it is impossible to identify a TrueCrypt container or partition.

TrueCrypt volume format version 1 specification:

Offset (bytes)	Size (bytes)	Encryption Status	Description
0	64	Not Encrypted	Salt *
64	4	Encrypted	Text string “TRUE”
68	2	Encrypted	Volume version
70	2	Encrypted	Minimum program version required to open the volume
72	4	Encrypted	CRC-32 checksum of the (decrypted) bytes 256-511
76	8	Encrypted	Volume creation time
84	8	Encrypted	Header creation/modification time
92	164	Encrypted	Reserved
256	32	Encrypted	Data used to generate IV and ‘whitening’ values
288	224	Encrypted	Master encryption key
512	N/A	Encrypted	Data area (the actual volume contents)

\*) Note that salt does *not* need to be encrypted, as it does not have to be kept secret (salt is a sequence of random values).

The bytes 0-63 (salt), bytes 256-287 (data used to generate IV and ‘whitening’ values), and bytes 288-511 (master encryption key), contain random values that have been generated using the built-in random number generator (see section *Random Number Generator*) during the volume creation process.

## Header Key Derivation Function

Header key is used to decrypt the encrypted area of the TrueCrypt volume header (see sections *Encryption Scheme* and *TrueCrypt Volume Format Specification*). The technique that TrueCrypt uses to generate header keys conforms to PKCS #5 v2.0 (see <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf>). A 64-byte (512-bit) salt is used, which means there are 2 to the power of 512 keys for each password. This significantly decreases vulnerability to ‘off-line’ dictionary attacks (precomputing all the keys for a dictionary of passwords is very difficult when a salt is used). 2,000 iterations of the key derivation function have to be performed to derive a header key, which significantly increases the time necessary to perform an exhaustive search for passwords. The header key derivation function is based on HMAC-SHA-1 (for more information, please see RFC 2104, available at: <http://www.cis.ohio-state.edu/htbin/rfc/rfc2104.html>).

## Sector Scrambling

Each cipher implemented in TrueCrypt operates in CBC mode (Triple-DES in inner-CBC). The IV (initialization vector) is a random value, which is unique to each sector and volume. This value is generated as follows:

1. Bytes 256-263 of the decrypted volume header are retrieved (see sections *TrueCrypt Volume Format Specification* and *Encryption Scheme*).
2. Data retrieved in (2) are XORed with the 64-bit sector number (each sector is 512 bytes long; sectors are numbered starting at 0).

Note: Step (1) is only performed once, right after the volume is mounted. The retrieved value remains in RAM then.

Every 8 bytes of each sector (after the sector is encrypted) are XORed with a 64-bit value, which is unique to each sector and volume. The value is generated as follows:

1. Bytes 264-271 of the decrypted volume header are retrieved (see sections *TrueCrypt Volume Format Specification* and *Encryption Scheme*).
2. Bytes 272-279 of the decrypted volume header are retrieved.
3. Data retrieved in (1) are XORed with the 64-bit sector number (each sector is 512 bytes long; sectors are numbered starting at 0).
4. Data retrieved in (2) are XORed with the 64-bit sector number.
5. A 32-bit CRC-32 value of the first 8 bytes of the resultant value in (3) is calculated.
6. A 32-bit CRC-32 value of the second 8 bytes of the resultant value in (3) is calculated.
7. A 32-bit CRC-32 value of the first 8 bytes of the resultant value in (4) is calculated.
8. A 32-bit CRC-32 value of the second 8 bytes of the resultant value in (4) is calculated.
9. The value calculated in (5) is XORed with the value calculated in (8).
10. The value calculated in (6) is XORed with the value calculated in (7).
11. The 32-bit value calculated in (9) is written to the upper 32-bit word and the value calculated in (10) is written to the lower 32-bit word of the 64-bit 'whitening' value.

## Random Number Generator

The random number generator implemented in TrueCrypt is used to generate the salt, master encryption key, IV and 'whitening' values.

The random number generator creates a pool of random values in RAM. The pool, which is 256 bytes long, is periodically filled byte by byte with values derived from the following sources:

- ! Mouse movements (CRC32-hashed coordinates and event delta times)
- ! Mouse clicks (CRC32-hashed event delta times)
- ! Key presses (CRC32-hashed key codes and event delta times)
- ! Network interface statistics (NETAPI32) (collected only once in the beginning)
- ! Performance statistics of disk devices (collected only once in the beginning)
- ! Various Win32 handles, time variables, and counters (collected at 250-ms interval)

Random values are written to the pool by adding (not by replacing the old values in the pool). This means that from the moment that a value is written to the pool, it never stops affecting the state of it. Additionally, after a value (byte) is added to the pool, the pool is entirely hashed using a hash function (SHA-1).

The described random number generation technique is based on the following:

- ! *Software Generation of Practically Strong Random Numbers* by Peter Gutmann, (<http://www.cs.auckland.ac.nz/~pgut001/pubs/random.pdf>)
- ! *Cryptographic Random Numbers* by Carl Ellison, (<http://www.clark.net/pub/cme/P1363/ranno.html>)

## Compliance with Standards

TrueCrypt complies with the following standards and specifications:

- ! PKCS #5 v2.0 (<ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf>)
- ! FIPS 180-2 – SHA-1 ([http://csrc.nist.gov/publications/fips/fips180-2/FIPS180-2\\_changenotice.pdf](http://csrc.nist.gov/publications/fips/fips180-2/FIPS180-2_changenotice.pdf))
- ! RFC 2104 – HMAC (<http://www.cis.ohio-state.edu/htbin/rfc/rfc2104.html>)
- ! RFC 2202 – HMAC-SHA-1 (<http://www.cis.ohio-state.edu/htbin/rfc/rfc2202.html>)
- ! FIPS 46-3 – Triple-DES (<http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>)

## TrueCrypt System Files

`WindowsPath\TrueCryptSetup.exe` (uninstaller)

### Windows 2000/XP

`WindowsPath\SYSTEM32\DRIVERS\truecrypt.sys` (driver)

`WindowsPath\SYSTEM32\TrueCryptService.exe` (service)

The service is necessary to register each newly mounted volume with the operating system (Mount Manager), which then assigns a drive letter to it. The service also allows users without administrator privileges to dismount TrueCrypt volumes.

## Windows 98/ME

*WindowsPath*\SYSTEM\IOSUBSYS\truecrypt.vxd (driver)

Note: Replace *WindowsPath* with your Windows installation path (e.g. C:\WINDOWS)

## Known Bugs and Device Driver Limitations

### Windows 2000/XP

- ! Network volumes are not supported.
- ! Raw floppy disk volumes: when a floppy disk is ejected and another one is inserted, garbage will be read/written to the disk, which could lead to data corruption. This affects only raw floppy disk volumes (not containers stored on floppy disks).

### Windows 98/ME

- ! The driver is single-threaded. Under Windows 2000/XP it is possible to store one TrueCrypt volume *within* another, and mount both volumes (since the Windows 2000/XP driver is multi-threaded). This is *not* possible under Windows 98/ME and should not be attempted.
- ! Raw floppy disk volumes are not supported.
- ! TrueCrypt volumes must be dismounted manually.
- ! Up to eight TrueCrypt volumes can be mounted at a time.
- ! The driver incorrectly allows the OS to 'believe' that the very first write to a read-only volume (i.e. a floppy disk with the write protection on) has been successful. This, for example, makes the OS consider a file updated, when it actually is not. Upon a subsequent write, the OS correctly reports the volume as write protected.

If you find a bug, please send us a bug report at [bugs@truecrypt.org](mailto:bugs@truecrypt.org) and if possible, include the following information:

- ! Name and version of the operating system
- ! Service pack version, if applicable
- ! TrueCrypt version
- ! Encryption algorithm
- ! TrueCrypt volume size and type (whether it is a device, partition, or a file-hosted volume)
- ! What applications were running and what was going on at the time when the bug occurred
- ! Full error report

# Frequently Asked Questions

For the latest version of TrueCrypt FAQ please see: [www.truecrypt.org/faq.html](http://www.truecrypt.org/faq.html)

## Version History

### 1.0

February 2, 2004

*Note: TrueCrypt is based on E4M (Encryption for the Masses). Therefore, the following list contains differences between E4M 2.02a and TrueCrypt 1.0 (minor differences have been omitted).*

#### Improvements:

- ! Windows XP/2000 support
- ! The maximum volume size is 18,446,744,073 GB (E4M only allows 2 GB).  
Note: File system, hardware connection standard, and operating system limitations have to be taken into account when determining maximum volume size.
- ! Plausible deniability. It is impossible to identify a TrueCrypt container or partition. Until decrypted, a TrueCrypt volume appears to consist of nothing more than random data (it does not contain any "signature"). Therefore, it is impossible to prove that a file, a partition or a device is a TrueCrypt volume and/or that it has been encrypted. To achieve plausible deniability, the format of the volume and the encryption process had to be significantly changed.
- ! The salt is 64 bytes long now (E4M uses 20 bytes).
- ! The iteration count of the key derivation function increased to 2,000 (E4M uses 1,000).
- ! Free space is filled with random data during volume creation, instead of filling it with zeroes. This reduces the amount of predictable plaintext and, in future, will increase the level of plausible deniability of hidden volumes.
- ! Up to 32 partitions per physical disk drive can be encrypted now (Windows XP/2000).
- ! The minimum volume password length has been increased to 12 characters.
- ! The maximum volume password length has been decreased from 100 to 64 characters. This was necessary to avoid the following: When a password longer than 64 characters was passed to HMAC-SHA-1, the whole password was first hashed using SHA-1 and the resultant 160-bit value was then used instead of the original password (which complies with HMAC-SHA-1 specification), thus the password length was in fact reduced.
- ! The Blowfish key length size has been increased to 448 bits.  
Remark: Even though our increasing the key size to 448 bits might appear to be a significant overkill, there was no reason for us not to do so (note that there is no decrease in speed of encryption/decryption).



**Bug fixes:**

- ! Sector scrambling algorithm flaw fixed. Two or more disk sectors to be encrypted consisting of the same values (e.g. filled with zeroes), after being encrypted by E4M, start with the same 8-byte sequence of values (i.e. the first eight bytes of any of these encrypted sectors contain the same values as the first eight bytes of any other of these encrypted sectors). If this had not been fixed, the plausible deniability would not have been possible.
- ! TrueCrypt volumes can be dismounted (Windows XP issue).
- ! "Blue screen" errors no longer occur during Windows shutdown when there is one or more mounted TrueCrypt volumes.
- ! Drive geometry is calculated correctly now (*chkdsk.exe* and *format.exe* do not fail anymore).
- ! A TrueCrypt volume can be reformatted as FAT32 or NTFS using the Windows built-in format tool (Windows XP/2000 issue).
- ! Windows Check Disk can now be used on TrueCrypt volumes (Windows XP/2000 issue).
- ! Windows Disk Defragmenter can now be used on encrypted volumes (Windows XP/2000 issue).

**New features:**

- ! New IV (initialization vector) generation algorithm (see the documentation for more information)
- ! Every 8 bytes of each sector (after the sector is encrypted) are XORed with a random 64-bit value, which is unique to each sector and volume (sector is 512 bytes long). This makes obtaining a plaintext/ciphertext pair a bit more difficult.
- ! New function to clear the volume history.
- ! When selecting a partition/device, the sizes and file system types of available partitions/devices are displayed (Windows XP/2000).
- ! List of mounted TrueCrypt volumes now contains their sizes and encryption algorithms used (Windows XP/2000).
- ! Free volume space is reported (in 'My Computer' list etc.)
- ! Windows XP format facilities do not support formatting volumes larger than 32 GB as FAT32. However, with TrueCrypt Volume Creation Wizard it is now possible to create FAT32 volumes larger than 32 GB.
- ! New function that allows multiple TrueCrypt partitions to be mounted provided that their correct password(s) has/have been entered (this includes the cached passwords, if there are any).
- ! Quick format (partitions/devices only)
- ! Cluster size selection (when creating new volumes)
- ! Volume properties can now be examined (encryption algorithm, volume creation time, last password change time etc.)
- ! New function to dismount all mounted TrueCrypt volumes.
- ! New command line options to dismount all mounted TrueCrypt volumes: /d and /dismountall

- ! HMAC-SHA1 and CRC-32 algorithm tests are now included in the self-test facility.
- ! Program menu and Preferences window added.
- ! Custom user interface fonts supported.
- ! Optionally, the TrueCrypt installer can now create System Restore points (Windows XP/ME).
- ! Password input field is wiped after a correct volume password has been entered.
- ! New graphics, icons, user interface
- ! New documentation

**Removed features:**

- ! E4M and SFS volumes are no longer supported.
- ! DES cipher removed.
- ! HMAC-MD5 removed (to be replaced by HMAC-RIPEMD-160).

# Acknowledgements

Thanks to:

- ! Paul Le Roux for making his E4M source code available. TrueCrypt is based on E4M. For information on differences between E4M and TrueCrypt, please see *Version History*.
- ! Eric Young for writing his excellent libdes, libcast etc., which were the sources of some of the cryptography code used in E4M.
- ! 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 used in E4M.
- ! Andy Neville for providing some of the code and inspiration, useful in the implementation of the file-hosted volumes (E4M).
- ! Shaun Hollingworth (aka Aman), who allowed Paul Le Roux to develop the E4M driver for Windows 98/ME using the Scramdisk source code.
- ! David Kelvin, who added the privacy password command line argument, and the quiet mode.

# Contact

TrueCrypt team can be contacted at:  
mail@truecrypt.org

For more information on ways to contact us, please see:  
[www.truecrypt.org/contact.html](http://www.truecrypt.org/contact.html)