# TurboCrypt User Manual

First published: July 2008

Global IP Telecommunications, Ltd. & PMC Ciphers, Inc. - Josephsburgstr. 85, 81673 Munich, Germany
Tel. +49 89 235 1468-0

# TurboCrypt

## User Manual

For the latest information, please see http://www.pmc-ciphers.com

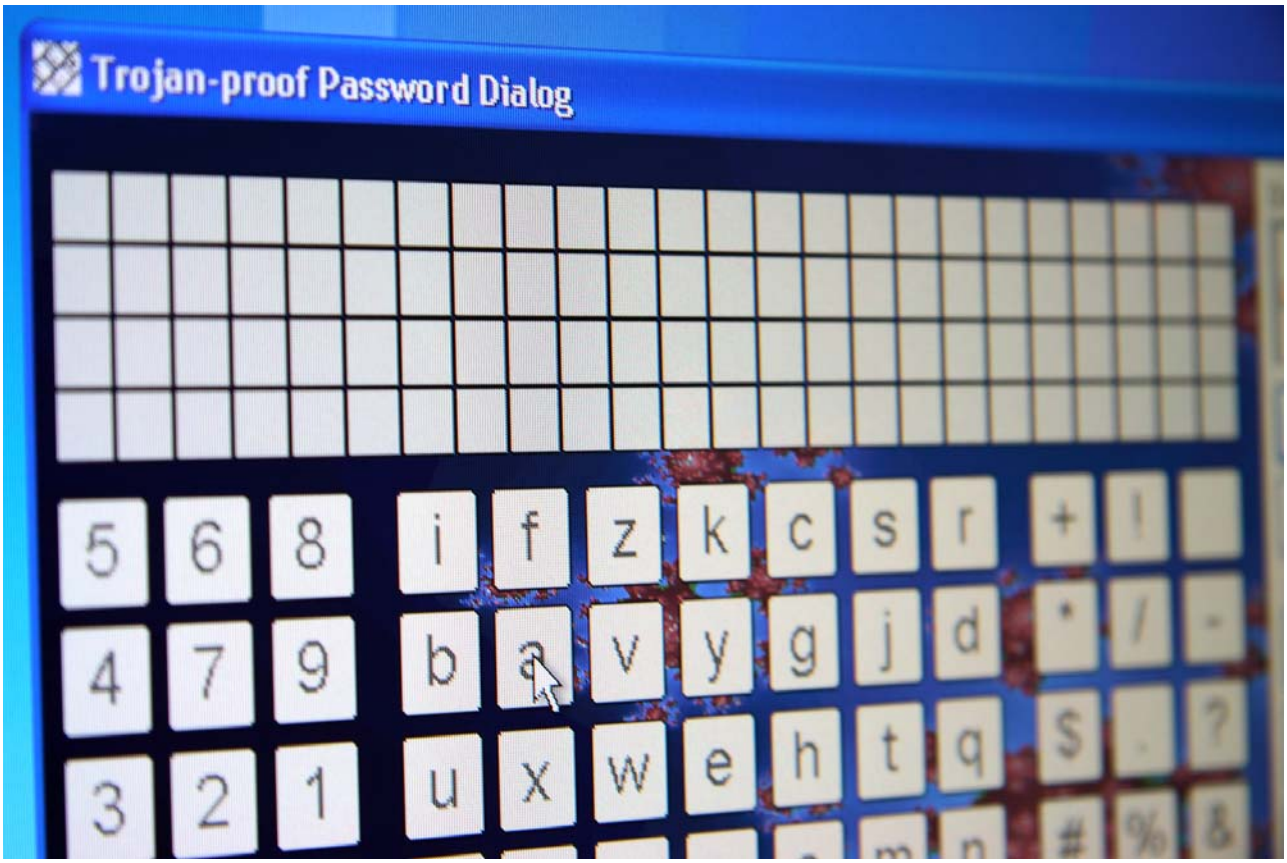
---

## Index of contents

## Introduction

The simple password protection mechanisms of popular Office packages or compression utilities can generally be broken or bypassed easily. Additionally, it can be assumed that all conventional protection mechanisms that are already built into modern Operating Systems can be bypassed by State Authorities.

In order to counteract, TurboCrypt provides the user with one or more encrypted volumes that can be securely accessed by specifying a user-selected password. Even if a number of Trojan Horses (i.e. malicious computer viruses that infect a user's computer and that report all keystrokes and/or screen content back to the server of a criminal or an intelligence agency) have infected the user's computer, users can still be sure that their password remains completely secret.

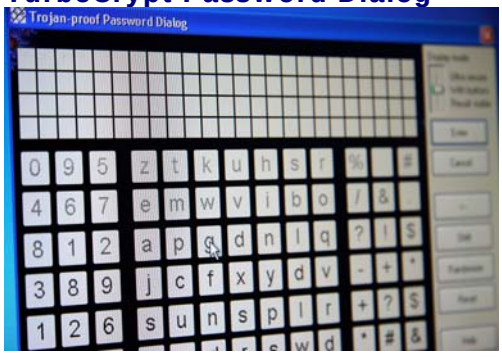



TurboCrypt is the very first product of its kind featuring a trojan horse-proof password dialog


The TurboCrypt encryption driver is capable of providing an additional disk drive to the user. Data that is written to or read from an encrypted TurboCrypt volume is actually read or written to a physical disk device like an internal hard disk, an external disk drive or a USB stick.

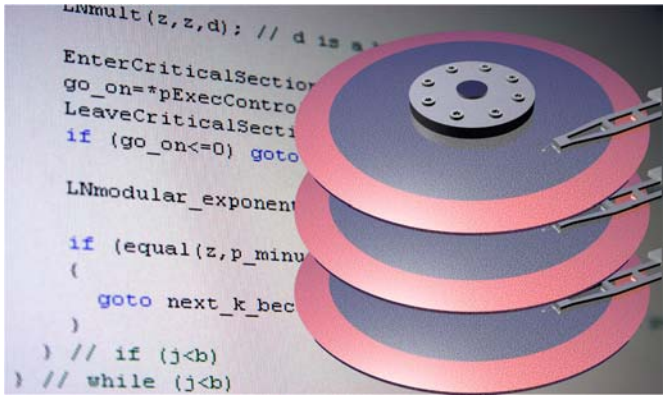
**User who wants to store data on an ultra-secure TurboCrypt volume**

**TurboCrypt Password Dialog**




User enters password which acts as encryption key

Data that is stored in an encrypted TurboCrypt volume is encrypted by the ultra-secure TurboCrypt encryption driver



**TurboCrypt Driver**

All TurboCrypt data is read or written from/to volume image files. Data is ALWAYS encrypted, even if a power outage occurs.

**Internal or external hard disk or USB memory stick**

*Back to Index of contents*

## Installation

The 32 bit version of TurboCrypt installs on computers with Microsoft Windows XP (32 bit) and Microsoft Windows Vista (32 bit). The 64 bit version installs on 64 bit Windows Vista. Simply launch the setup program and follow the instructions on the screen.

**If you're using Windows Vista, you MUST launch TurboCrypt with administrative rights! Otherwise the software would not be able to install the encryption driver.**
**TurboCrypt is programmed to request the execution level of an Administrator. You're automatically asked by the Operating System if you want to launch TurboCrypt with Administrator execution level.**

When TurboCrypt is launched for the first time, the user interface will show up as below:


Please select two drive letters that will be used by TurboCrypt permanently. When clicking at the  (OK) button, TurboCrypt will install two instances of a software driver that make available two extra drives with your previously selected drive letters.
The drive letters will appear in Windows Explorer only AFTER rebooting the computer.

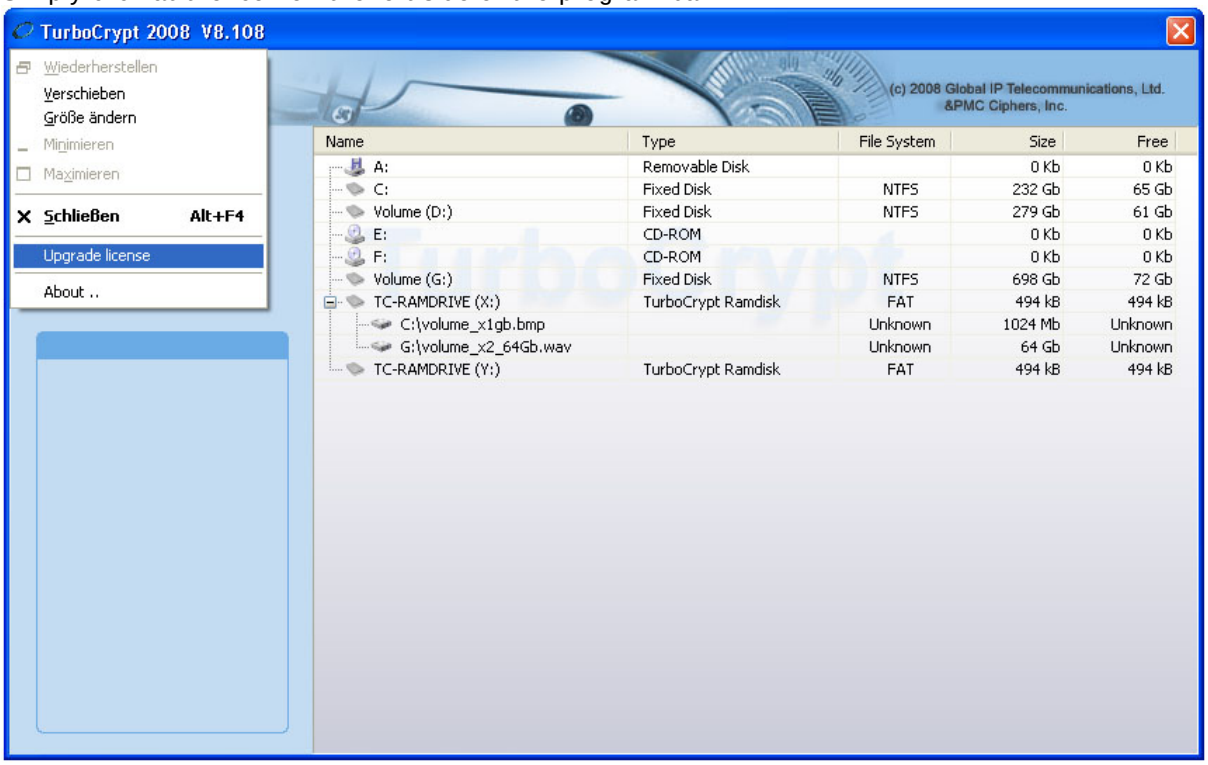
By default each TurboCrypt encryption driver creates a RAM disk drive of approx. 512Kb drive size. This Ramdisk can be used like any other disk, but the contents of the disk will be lost when the machine is shutdown. Any TurboCrypt encrypted volume can be mounted by the encryption driver to the driver's drive letter. TurboCrypt drives thus behave exactly like memory card readers or floppy disk drives.

*Back to Index of contents*

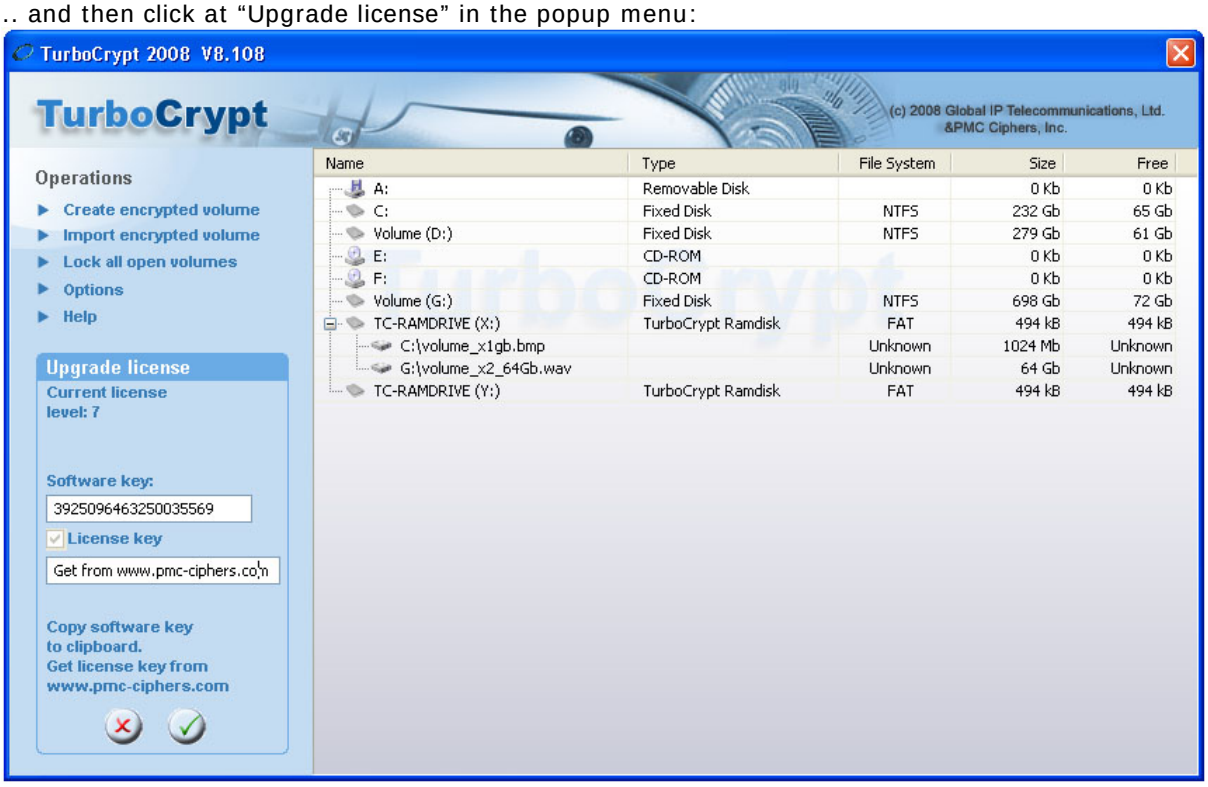
## Registering TurboCrypt

In order to be able to create large encrypted volumes and to take advantage of the full functionality provided by TurboCrypt you can get a license key from www.pmc-ciphers.com. The license key is free up to 1Gb volume size. **You can thus use the full functionality of TurboCrypt FOR FREE !!!**

Simply click at the icon on the left side of the program bar:




.. and then click at “Upgrade license” in the popup menu:



The blue box on the left side displays all required information:

| | |
|---|---|
| Software key: | Please copy the full string (without changing it) to the clipboard. Proceed to http://www.pmc-ciphers.com, open an account in the shop system and paste the software key there. |
| License key: | Paste the key from the shop system at www.pmc-ciphers.com in this control and click at the OK button to activate the license. |

Click at the  (OK) button to activate the new license.

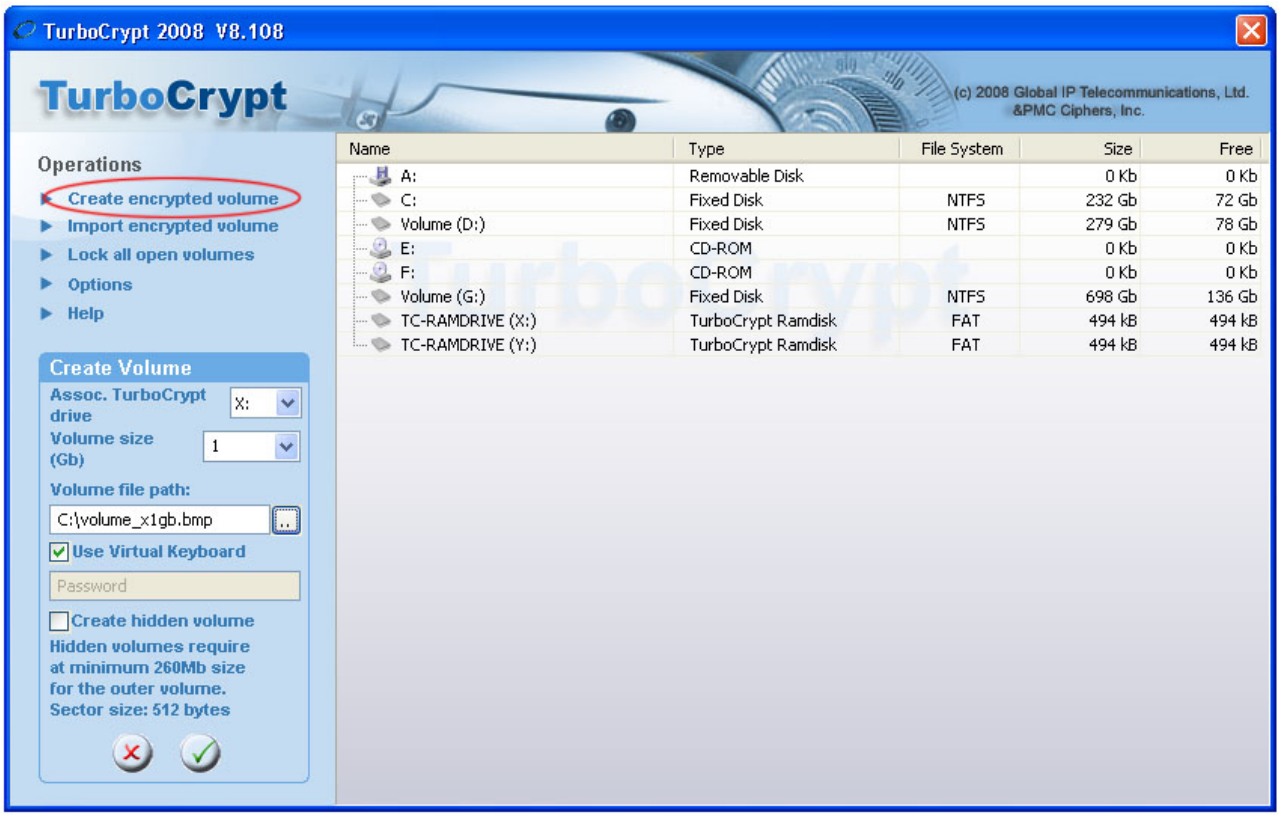
**If an erroneous license key is loaded, the software might react strangely. In this case you should first reset the license engine by entering the value -1 in the "License key" edit control, click at the  (OK) button, restart the software and then retry the process with a valid license key.**

The actual license level is always displayed when clicking at the icon on the left side of the program bar. 0 is the initial license level. Any level below 0 and above 8 indicates an invalid license.

*Back to Index of contents*

## Creation of an encrypted volume

TurboCrypt can manage almost 1000 different encrypted volumes that are simultaneously present.
An encrypted volume is either created or imported into TurboCrypt. In order to create an encrypted volume, click at “Create encrypted volume”:




The blue box on the left side displays all required controls:

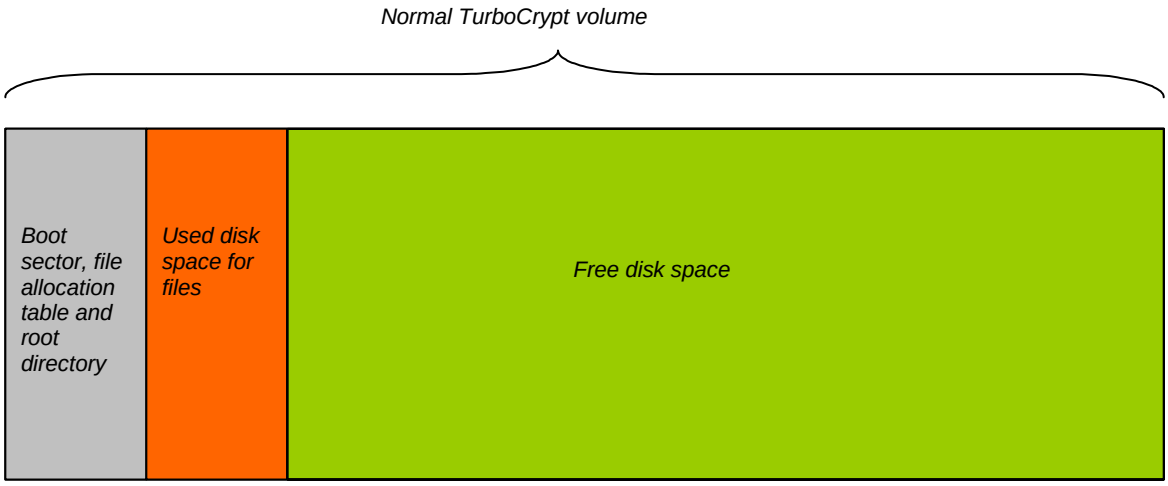
| | |
|---|---|
| Associated TurboCrypt drive: | Please select drive to which the new volume is to be assigned. |
| Volume size: | Total size of volume in gigabyte (1024 * 1024 * 1024 byte = 1024 Mb = 1.073.741.824 byte) |
| Volume file path: | Full file path to the volume file (image file). Creation of volume image files in the root directory of drive c: or in other protected directories might require administrative rights. Please launch TurboCrypt with the required rights (run as administrator) on Windows Vista. |
| Use Virtual Keyboard: | If this checkbox is in the checked state, you’ll be asked to enter the password in the trojan-horse-proof password entry dialog. If the checkbox is not in the checked state, please enter the password in the edit control below the checkbox. Please proceed to the explanation of the trojan-horse-proof virtual keyboard. |
| Create hidden volume: | If this checkbox is set to the checked state and if the selected volume size is greater or equal approx. 0.256Gb, you’ll be asked to slupply two additional parameters that are necessary to create a hidden and highly secret encrypted volume within the (outer) encrypted volume: |
| Sector 0: | Start sector of the hidden volume (please read the passage “Deniable volumes and deniable encryption” carefully !) |
| Password of hidden volume: | **Highly secret password** of the hidden and thus **DENIABLE** |

volume. **It is highly recommended to use the trojan-horse-proof virtual keyboard for password entry!!!**

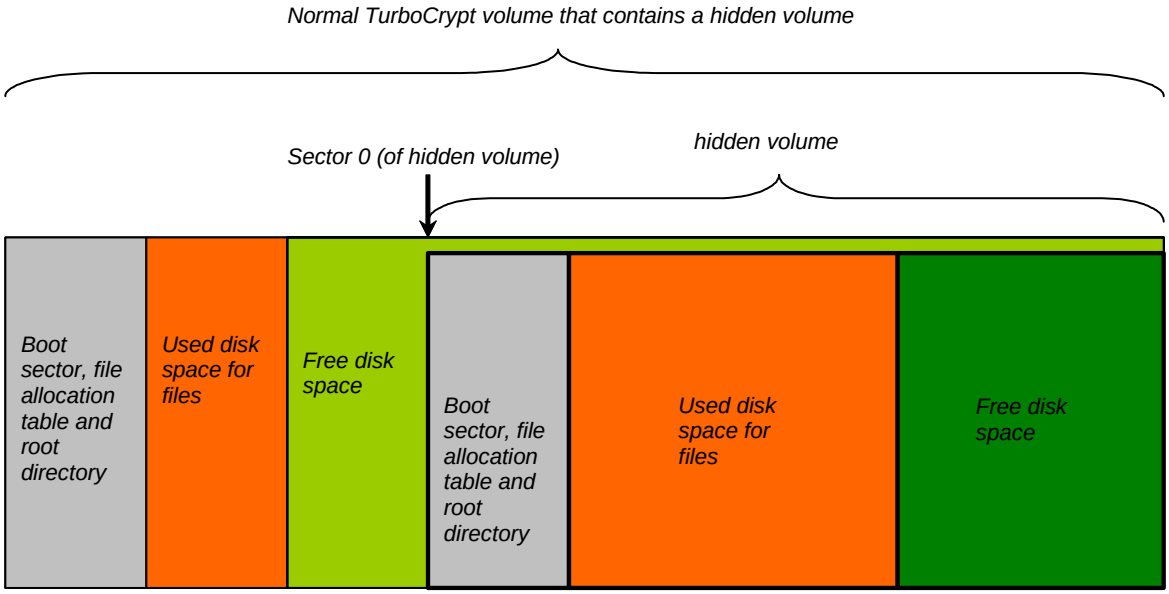
Click at the (OK) button to start the creation process. Creation of a volume can take minutes to hours, depending on volume size.

**Layout of hidden volumes:**

The start sector suggested by TurboCrypt is the first possible sector where a hidden volume can possibly start. Please choose either the suggested start sector or please select one that guarantees sufficient available memory for the files you plan to store in the outer (not deniable) volume. A sector is a group of 512 bytes. 2048 sectors thus correspond with 1Mb.



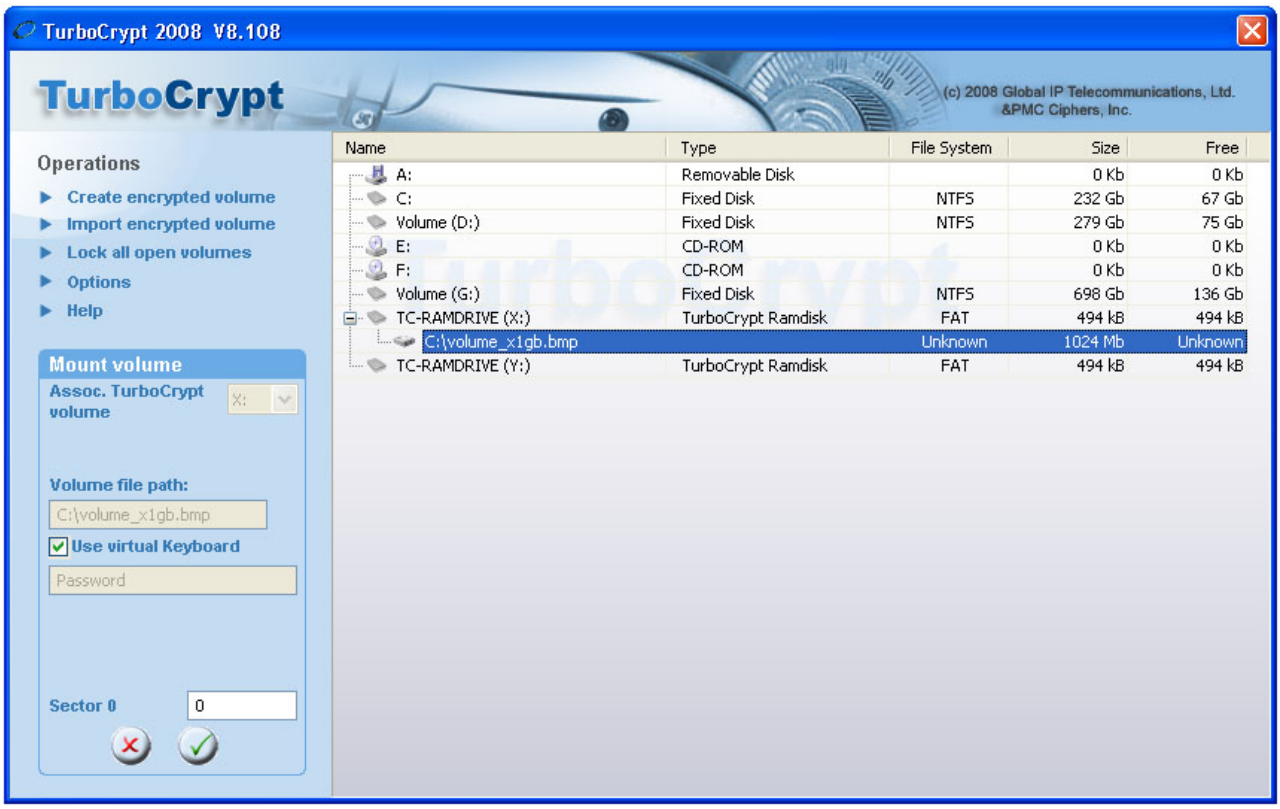

The following picture shows how a hidden volume is embedded in a normal TurboCrypt volume:




*Back to Index of contents*

## Mounting an encrypted volume

To each TurboCrypt drive one encrypted volume can be mounted at a time or the associated ramdisk is mounted alternatively. The ramdisk is always mounted when the operating system boots and when no encrypted volume is mounted.




In order to mount an encrypted virtual volume, click using the left mouse button on an encrypted volume in the tree view or click at the right mouse button and select “Mount volume” from the popup menu that will appear.

On the left side of the window in the blue box all required controls are displayed:

Associated TurboCrypt drive: TurboCrypt drive that is assigned to the selected volume (display only)

Volume file path: Full file path to the volume file (image file) (display only).

Use Virtual Keyboard: If this checkbox is in the checked state, you’ll be asked to enter the password in the trojan-horse-proof password entry dialog. If the checkbox is not in the checked state, please enter the password in the edit control below the checkbox. Please proceed to the explanation of the trojan-horse-proof virtual keyboard. **It is highly recommended to use the trojan-horse-proof virtual keyboard for password entry!!!**

Sector 0: Start sector of the hidden volume (please read the passage “Deniable volumes and deniable encryption” carefully !). If you want to mount a non-deniable volume, 0 is the correct start sector. **In case a hidden volume is within this outer volume, please be careful when adding files!!! You might overwrite the hidden volume!!! Loss of data would be the direct result!!!**
If you want to mount a hidden (deniable) volume, please enter the correct start sector for that hidden volume.

Password: Please enter password here in case you don't want to use the trojan-horse-proof virtual keyboard. **It is highly recommended to use the trojan-horse-proof virtual keyboard for password entry!!!**

Click at the  (OK) button to start the mount process.

The actual mount process consumes a substancial amount of processor time (CPU time) – depending on CPU speed more than 5 seconds.
TurboCrypt takes advantage of two unique mechanisms that increase attack security against the cipher as well as the underlying mount mechanism dramatically:

**1: High complexity of key setup (key expansion):**

The time required to convert the key for a cipher like AES (Advanced Encryption Standard) or DES (Data Encryption Standard) into so-called round keys requires only approx. 1µs. As a matter of consequence, it is very easy to try many different key combinations and thus to guess short passwords if AES, DES or a similar standard cipher is used. Such ciphers are generally used in conjunction with a compression function like SHA-256 or SHA-1, which either compress long passphrases into the key size of the selected cipher or which map short passphrases to the key size of the selected cipher. The time required to try a keys is the time required to compute the hash of the passphrase (approx. 0.3..3µs) plus the time to set up the round key (approx. 0.3 .. 3µs) plus the time to decrypt on sector (approx. 0.3 .. 3µs) yielding 0.9 .. 9µs in total. An attacker can thus potentially try one million different passphrases per second and thus potentially break a comparably short password.
The reason why most conventional ciphers feature this deficiency is simply the fact that fast generation of encryption context is usually seen as an advantage of small ciphers like DES, AES, Twofish, etc..
8 bit microcontroller hardware like the famous 8051 from Intel (1976), only tolerates small ciphers that compute round keys very quickly from the key. CPUs like the 8051 can be found in washing machines, toys, etc.. OTFE software like TurboCrypt although solely runs on CPUs like the Intel Pentium 4, Intel Core Duo, AMD Athlon64, etc.. Each of these target CPUs is a million times faster than an 8051, As a matter of consequence do modern microprocessors easily compute all round keys from a key (also called the crypto context) within a fraction of a microsecond.

TurboCrypt overcomes this deficiency of common OTFE software by taking advantage of the extremely high complexity of polymorphic ciphers. Instead of just 52 byte AES round key data, the 1024 bit Polymorphic Cipher implemented in TurboCrypt is 40kbyte in size. More than 100.000.000 operations are required to compute the crypto context of that cipher. As an adversory cannot take a shortcut, he is inevitably slowed down by factor 20.000 compared with AES.

**2: Secure password transport to the TurboCrypt encryption driver:**
States who develop their own viruses will definitely spend a few bucks on enabling their trojan horse(s) to spy on the driver stack. OTFE software always passes the password down to the encryption driver when a volume is to be mounted. If the password is transported in the clear, then there's no security at all. A well-programmed trojan-horse can intercept this data and as this is technically possible, it will be done.
As an example does the TurboCrypt competitor product named "TrueCrypt" use IO control code 466944 to signal a mount request to the encryption driver. This request is passed through the driver stack. Together with this mount request, the software passes the password used to open a specific encrypted volume in the clear through the stack. A trojan horse has not much more to do than to filter IRPs (IO Request Packets) for known IO control codes. It is obvious that this kind of weakness is disastrous, but it has nothing to do with the cipher itself.
**TurboCrypt encryption driver and control panel exchange vital information through an encryption protocol similar to SSL. Trojan horses can thus NOT get hold of password information.**


The encryption method used by TurboCrypt is the so-called Diffie-Hellman key exchange:

Diffie-Hellman key exchange uses modular exponentiation to yield a unique key that is only known to the two parties that exchange the key. In the following explanation it is assumed that the key exchange is initiated by the client and the client communicates with the driver. The client chooses a long integer number $a$ and calculates $\alpha$ using the following formula:

$\alpha = s^a \bmod p$ ;

$p$ is a fix and publically known long prime number
$s$ is a fix and publically known primitive root mod p
$a$ is freely chosen by the client. The client keeps $a$ secret.
$\alpha$ is the public result of the computation performed by the driver.

$\alpha$ is sent to the driver. $s$ and $p$ are known to the driver. The driver chooses a long integer number $b$ and computes $\beta$ using the following formula (same formula as above):

$\beta = s^b \bmod p$ ;

$p$ is a fix and publically known long prime number
$s$ is a fix and publically known primitive root mod p
$b$ is freely chosen by the client. The driver keeps $b$ secret.
$\beta$ is the public result of the computation performed by the client.

The driver performs another computation prior to completing the IRP:

$k = \alpha^b \bmod p$ ;

$k$ is the negotiated key. The driver keeps $b$ and $k$ secret.

The driver completes the IRP and sends $\beta$ to the client.

The client computes $k$ as well through the following formula:

$k = \beta^a \bmod p$ ; $k$ is the negotiated key. The client keeps $a$ and $k$ secret.

Both parties now share an information that is not accessible by malicious software sniffing driver communication.

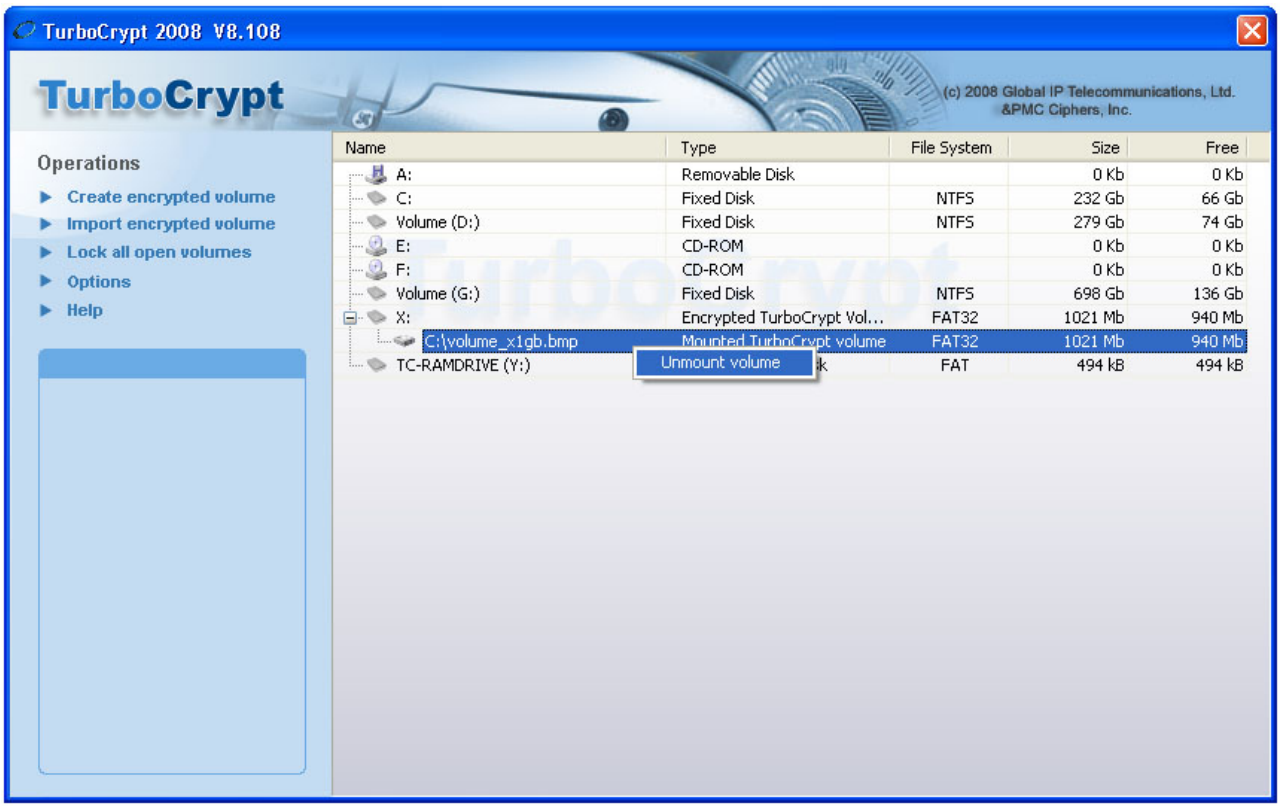
For the sake of completeness here's the proof that both $k$ are identical:

$$k = (s^b)^a \bmod p = (s^a)^b \bmod p$$

*Back to Index of contents*

## Unmounting (dismounting) an encrypted volume

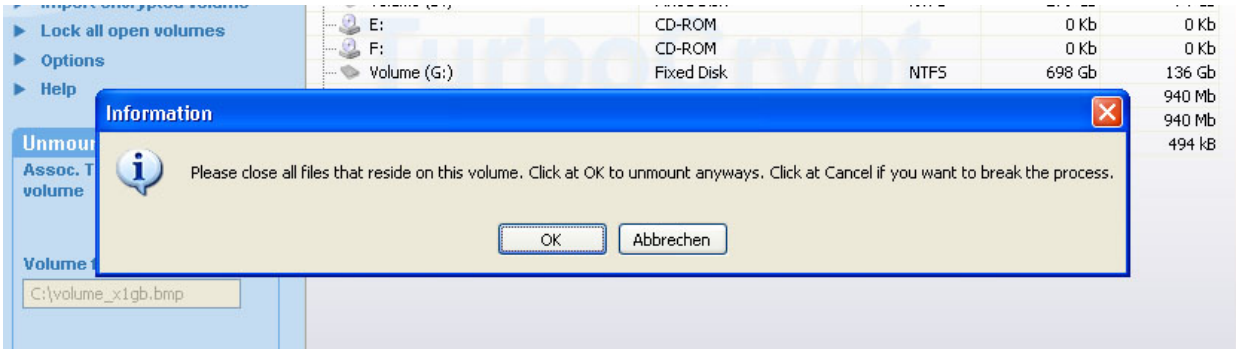
In order to dismount an encrypted volume, simply click at the mounted volume and select “Unmount volume”. Click at the (OK) button to start the dismount process.




Prior to dismounting an encrypted TurboCrypt volume or a ramdisk, TurboCrypt tries to lock the drive for exclusive access and then to eject the media.

TurboCrypt can only safely eject a removable media by first making sure that no files on the media are being used. Once this is done, TurboCrypt can eject the media and subsequently the driver can dismount and close the encrypted volume.

**If it is not possible for the operating system to eject the media, the root cause are typically open files. TurboCrypt asks the user in such cases if the volume shall be unmounted anyways (please see screenshot below). If the user clicks at “OK”, loss of data might occur.**




*Back to Index of contents*

## Creating a backup of an encrypted volume

OTFE (On-The-Fly-Encryption) software is generally susceptible to a ciphertext-only attack called "Backup Attack".

**TurboCrypt is the first software of its kind that provides the required functionality to guarantee immunity to this kind of attack.**

**Backup Attack:**

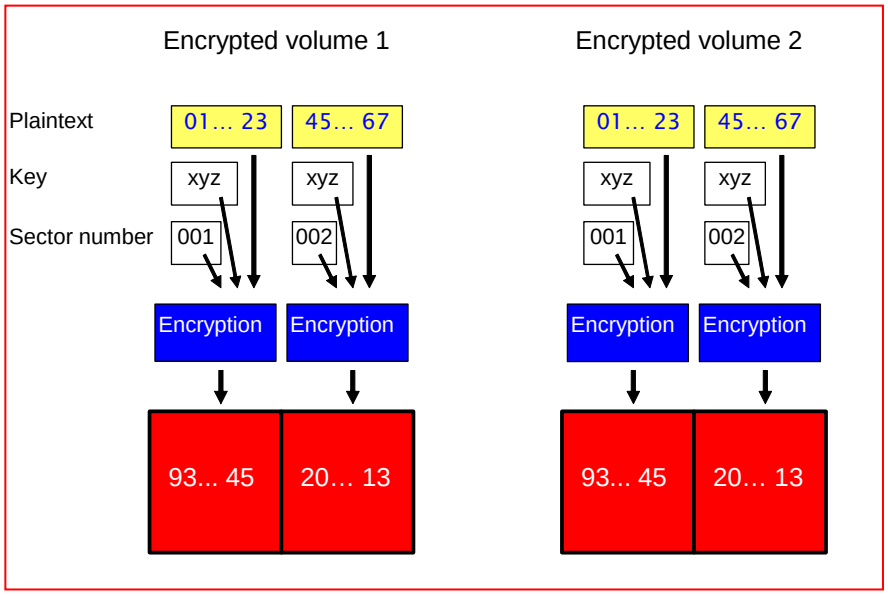
If a volume file is copied and the original copy is used to encrypt data while the other copy contains known plaintext (e.g. all zeros), it is possible to simply subtract data bits with identical bit positions in the two files from each other. This attack requires NO knowledge of the key used for encryption and it applies to ECB Mode (Electronic Codebook), Counter Mode (CM), Galois/Counter Mode (GCM), LRW, XEX, XTS, as well as CBC-based modes of disk encryption applications (OTFE).

It is very easy to unveil large parts of the sample image. All that is needed is the ciphertext of the sample image and the ciphertext of an image with a uniform color. White color was used to demonstrate the attack on the images below:

The two images are simply created by subtracting or multiplying the color of each pixel that is located at the very same position in the two ciphertext images.

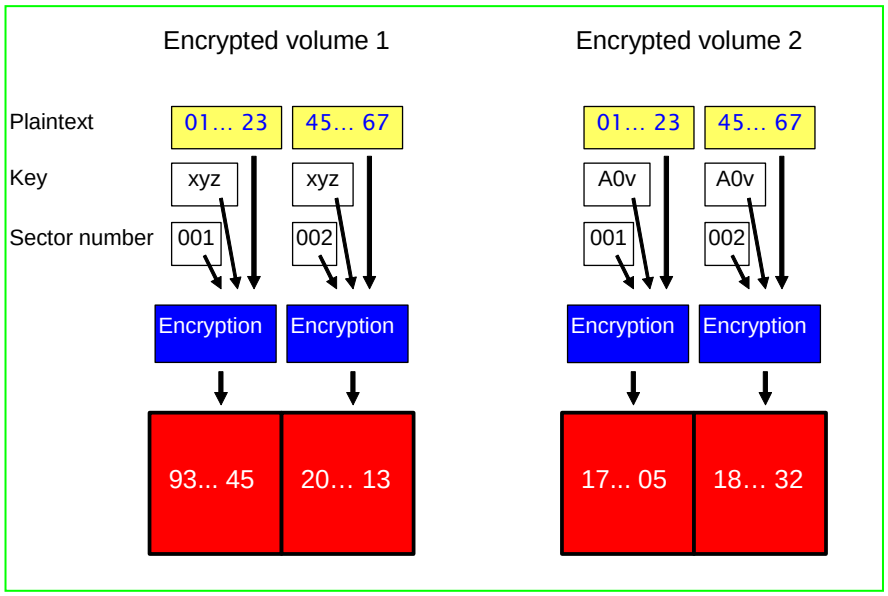
Encrypted image – encrypted image with all white pixels (subtraction)

Encrypted image multiplied with encrypted image with all white pixels




The result is the logical consequence of encrypting identical information with an identical key. It would be different if the user would have created another image file, instead of copying a volume file. When copying an encrypted volume, disk key and initialization vector information are also copied. This finally results in two identical keys that are used for both encrypted volumes.

Most or all OTFE software packages take advantage of disk keys. Changing passwords does thus not require re-encryption of an entire image file and security does not suffer at all due to the fact that password encryption is performed using a one-time-pad. The user-selected key serves as key for the encryption of the disk key, which is a true random number.

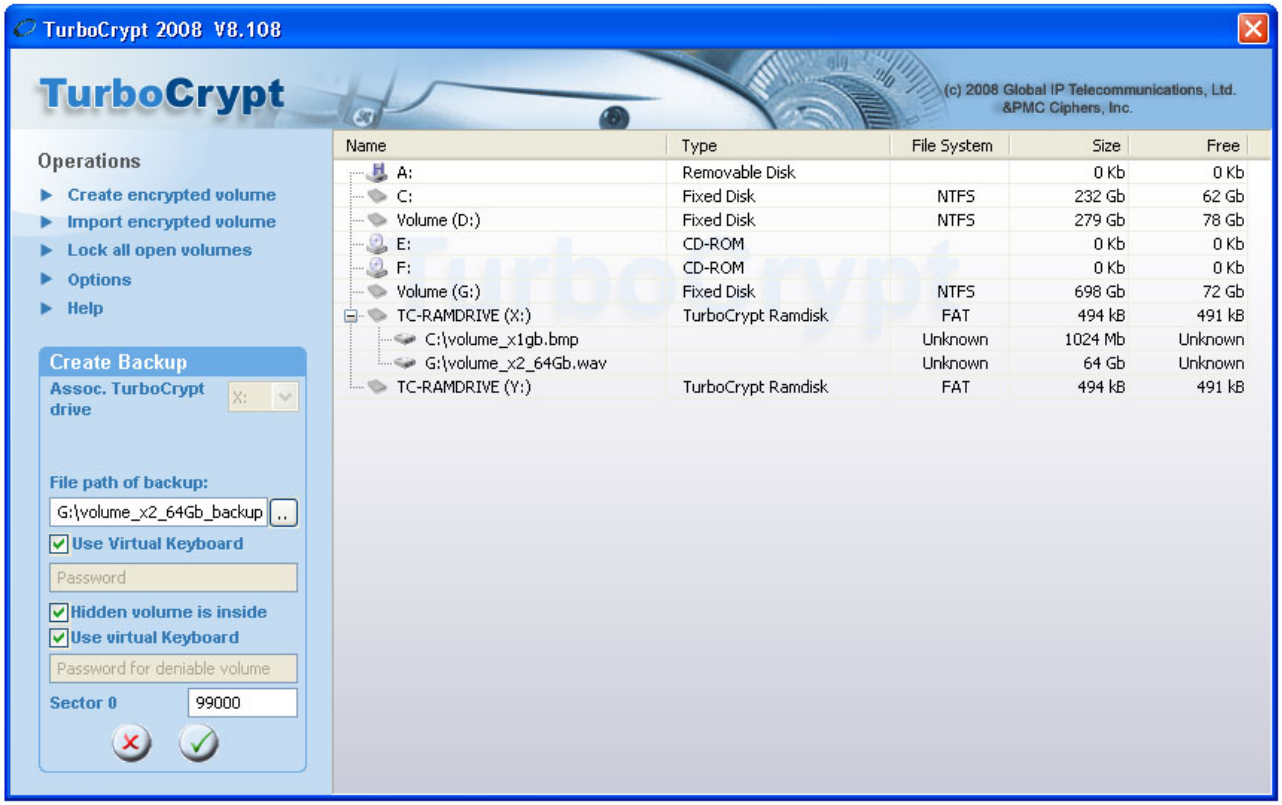



TurboCrypt creates backups of volume image files. **If users keep this rule in mind, 100% security can be guaranteed.**

TurboCrypt selects a new disk key for each backup. If disk keys differ between all copies of an image file, it is absolutely impossible to yield any information other than noise when comparing ciphertexts against each other.

When creating a backup of a volume image file, TurboCrypt uses a new real random key in place of the original disk key in the backup of an image file. This methodology solves the previously described security problem entirely.

As a matter of consequence, when TurboCrypt is given the command to create a backup, TurboCrypt needs to know the password of the outer volume as well as the inner (hidden) volume, if one is present. In order to create a backup, click at the volume file in the tree view and select “Create backup”. Subsequently specify one or two passwords (depending on presence of a hidden volume), as well as the correct start sector if a hidden volume is present.



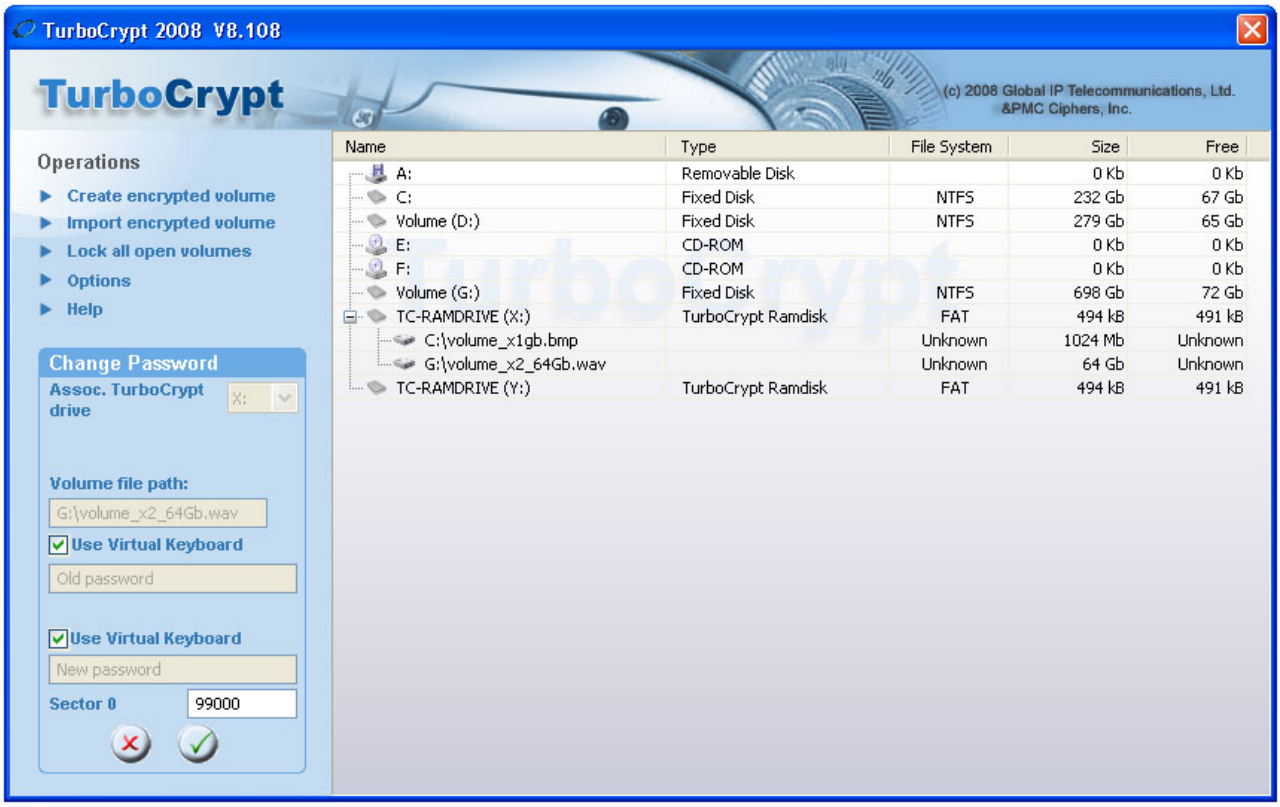

Click at the (OK) button to start creation of the backup. If all information you've provided is correct and if sufficient disk space is available, the backup will be created without any error message.

*Back to Index of contents*

## Change password of an encrypted volume

Like most OTFE software packages, TurboCrypt takes advantage of disk keys. Changing passwords is thus a matter of seconds even for very big volume files.
There is no need to re-encrypt the entire image file and security does not suffer at all due to the fact that password encryption is performed using a one-time-pad. The user-selected key serves as key for the encryption of the disk key, which is a true random number.

For changing the password of the outer volume or the (possibly present) hidden volume, simply click at an unmounted volume and select “Change password”. Click at the (OK) button to start the process. The example below shows how to change the password of a hidden volume starting at sector 99000. In the example the virtual keyboard is used both for entering the old password as well as the new password.

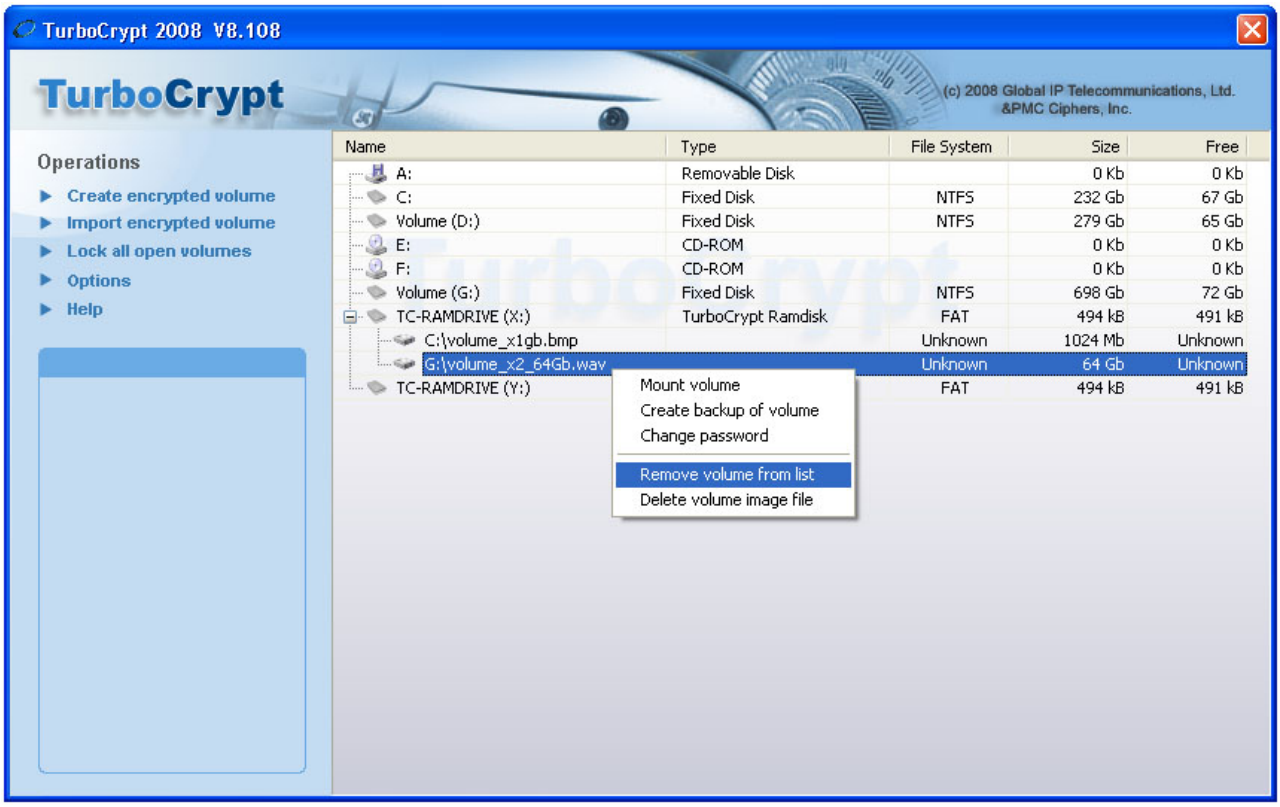



If you want to change the password of a normal or outer volume, please specify 0 for the start sector.

*Back to Index of contents*

## Remove volume from list / Delete volume

In order to only remove a specific volume from the list of available volumes, click at the volume file in the tree view using the right mouse button. Subsequently select “Remove volume from list” and confirm by clicking at the (OK) button located on the left side of the window in the blue box.



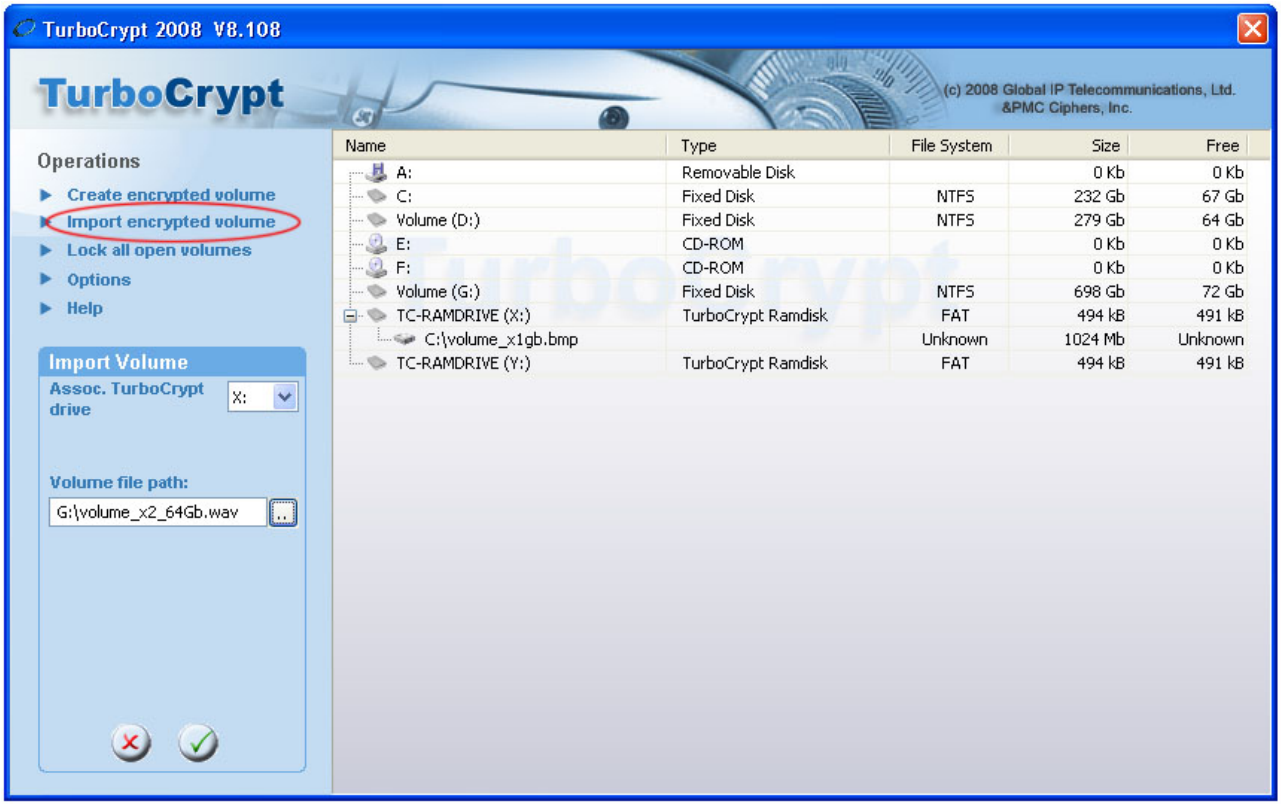

In order to delete a volume image file, click at “Delete volume image file” and confirm by clicking at the (OK) button.

Please note that deleted volume files might be recovered later only with specialized file recovery software. Such software is available as freeware from different manufacturers.

*Back to Index of contents*

## Importing an encrypted volume

Adding an already existing volume file to the list of volume files is easy using the "Import encrypted volume" command (on the left side of the main window).




On the left side of the window in the blue box all required controls are displayed:

Associated TurboCrypt drive: TurboCrypt drive that the imported volume will be assigned to.

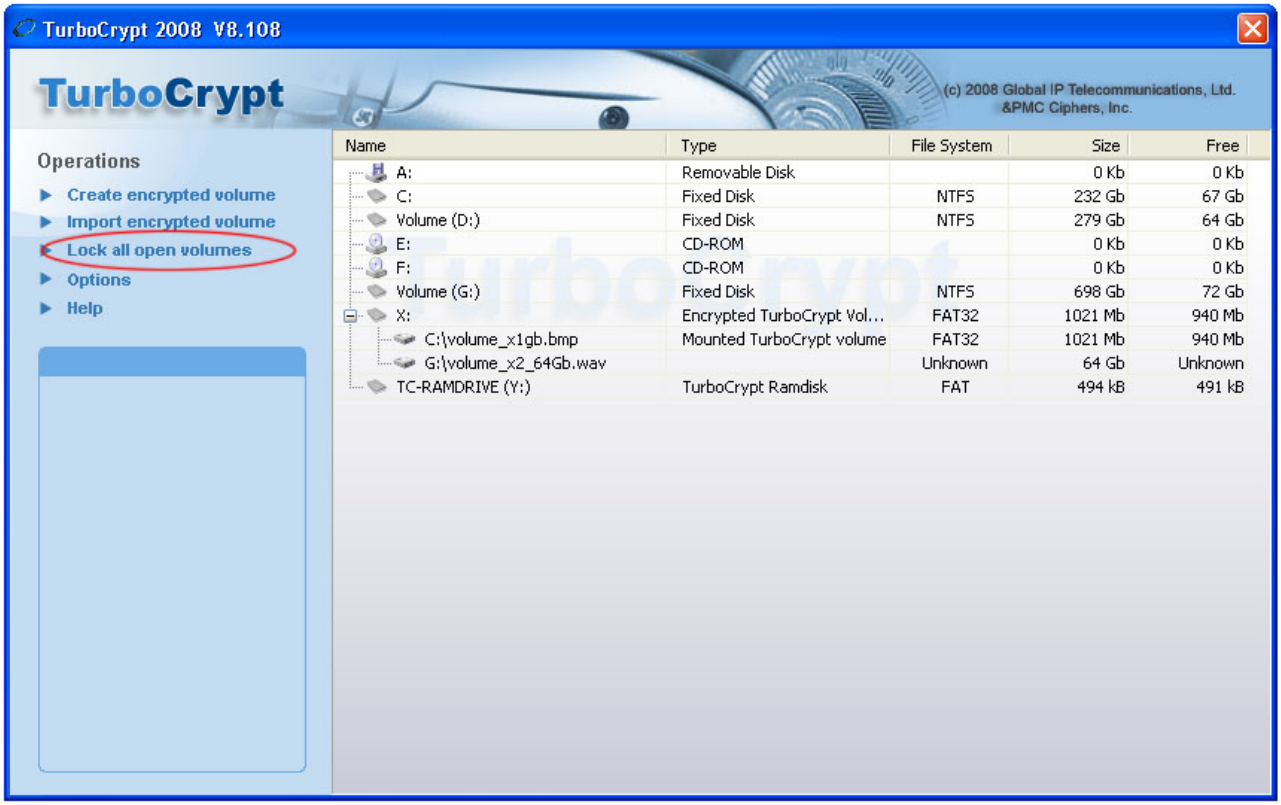
Volume file path: Full file path to the volume file (image file)

Click at the (OK) button to confirm import of the encrypted volume.

*Back to Index of contents*

## Lock all open volumes

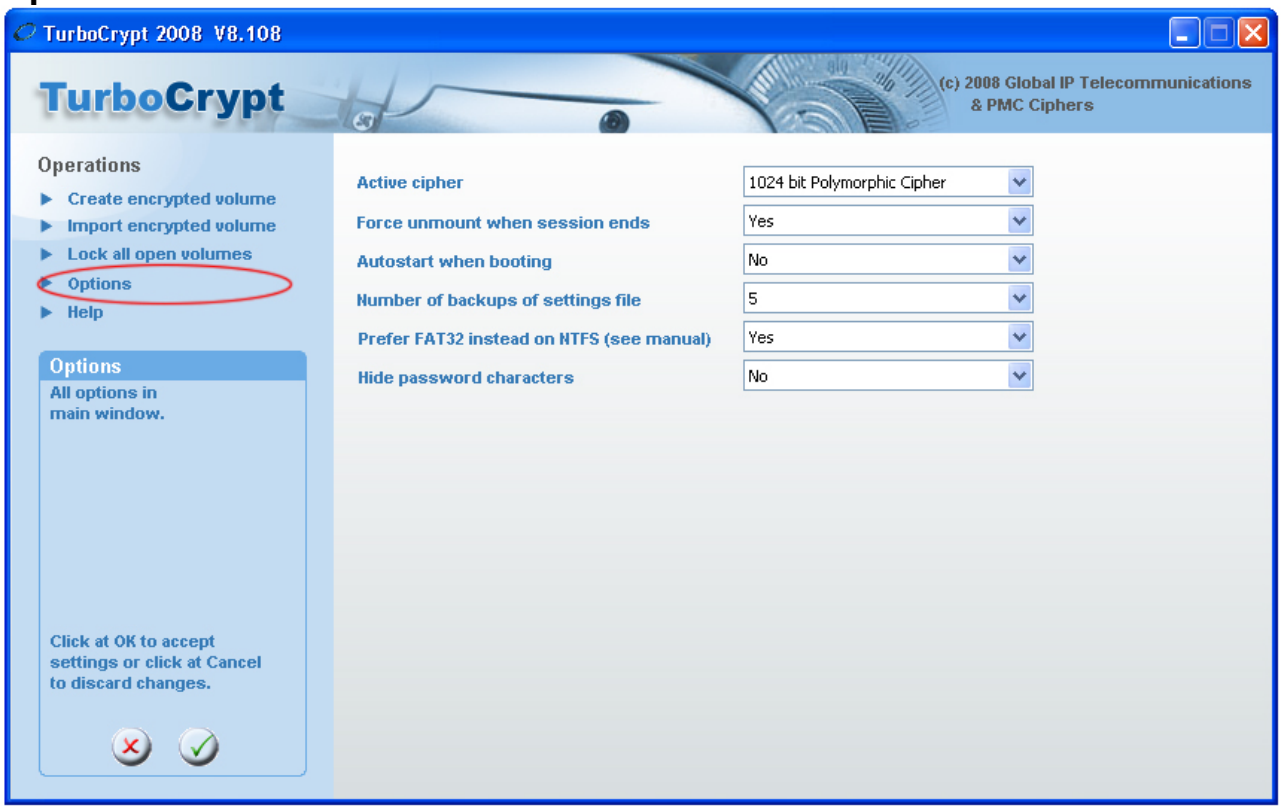
In order to dismount all open volumes at once, simply use the “Lock all open volumes” command (on the left side of the main window).




**IMPORTANT NOTE: This command ejects all open volumes and forces unconditional dismount! If files are still open, loss of data cannot be avoided in many cases. Users should use this command carefully!!!**

*Back to Index of contents*

## Options




The following program options are currently available:

Active cipher: 1024 bit Polymorphic Cipher or AES. It is highly recommended to use the 1024 bit Polymorphic Cipher as only this cipher has the potential to remain secure for the next 100 years.

Force unmount when session ends: If "Yes" is selected, the user interface will try to unmount all open mounted volumes when the user changes. The user interface needs to run in order to enable this functionality. The encryption driver itself is not notified by the operating system when a session ends.

Autostart when booting: If set to "Yes", TurboCrypt will be started automatically when the system boots and a user starts a session.

Number of backups of settings file: TurboCrypt saves all user settings in a text file located in the TurboCrypt installation folder. TurboCrypt creates a user-defined number of backups of this settings file (settings.txt). Experienced users can easily modify the settings.txt file if they want to.

Prefer FAT32 instead on NTFS (see manual): If set to "Yes", TurboCrypt will create volumes bigger than 4Gb as FAT32 formatted volumes rather than selecting automatically NTFS. If you change this option to "No", TurboCrypt will choose NTFS for large volumes. Speed is slower, but files can be bigger than 4Gb on NTFS-formatted volumes. The option is mainly required to guarantee true deniability for large volumes (>4Gb).

Hide password characters: If set to "Yes", TurboCrypt will show * characters instead of plaintext that is entered in all password edit controls (during volume creation, mount and creation of image file backups). **It is although highly recommended to use the trojan-horse-proof virtual keyboard instead of the keyboard of your**

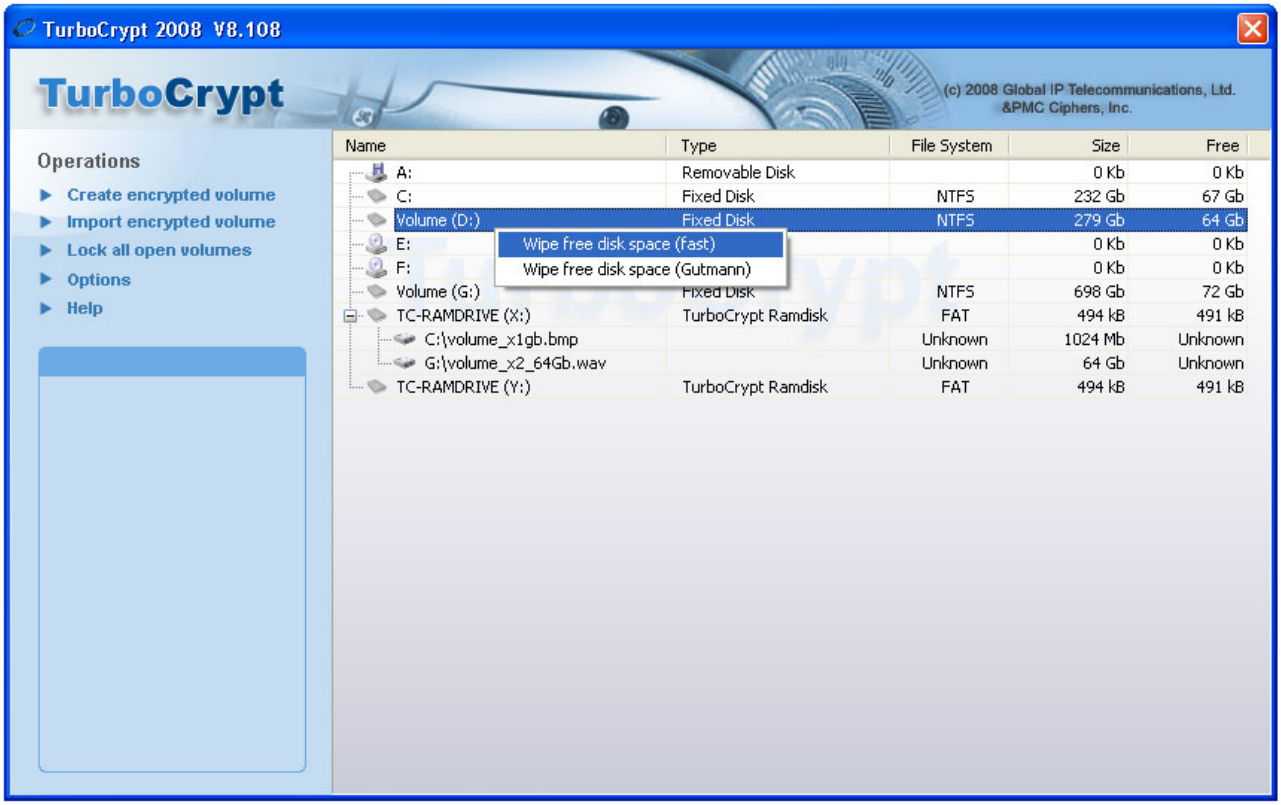
**computer! A keystroke logger could log everything that you type into the keyboard. This is by far more difficult if the trojan-horse-proof virtual keyboard is used!**

Click at the  (OK) button to confirm settings.

*Back to Index of contents*

## Wipe free disk space

For non-TurboCrypt drives with read/write capability, TurboCrypt makes available secure erase functionality of free disk space.
TurboCrypt supports fast wipe by writing pseudorandom data (biased by true randomness) in a file that occupies the entire free disk space of the selected drive.
TurboCrypt additionally makes available secure wipe using the Gutmann method. Gutmann wipe comprises 35 passes. During those 35 passes, a multitude of different bit patterns are written to the free disk space.

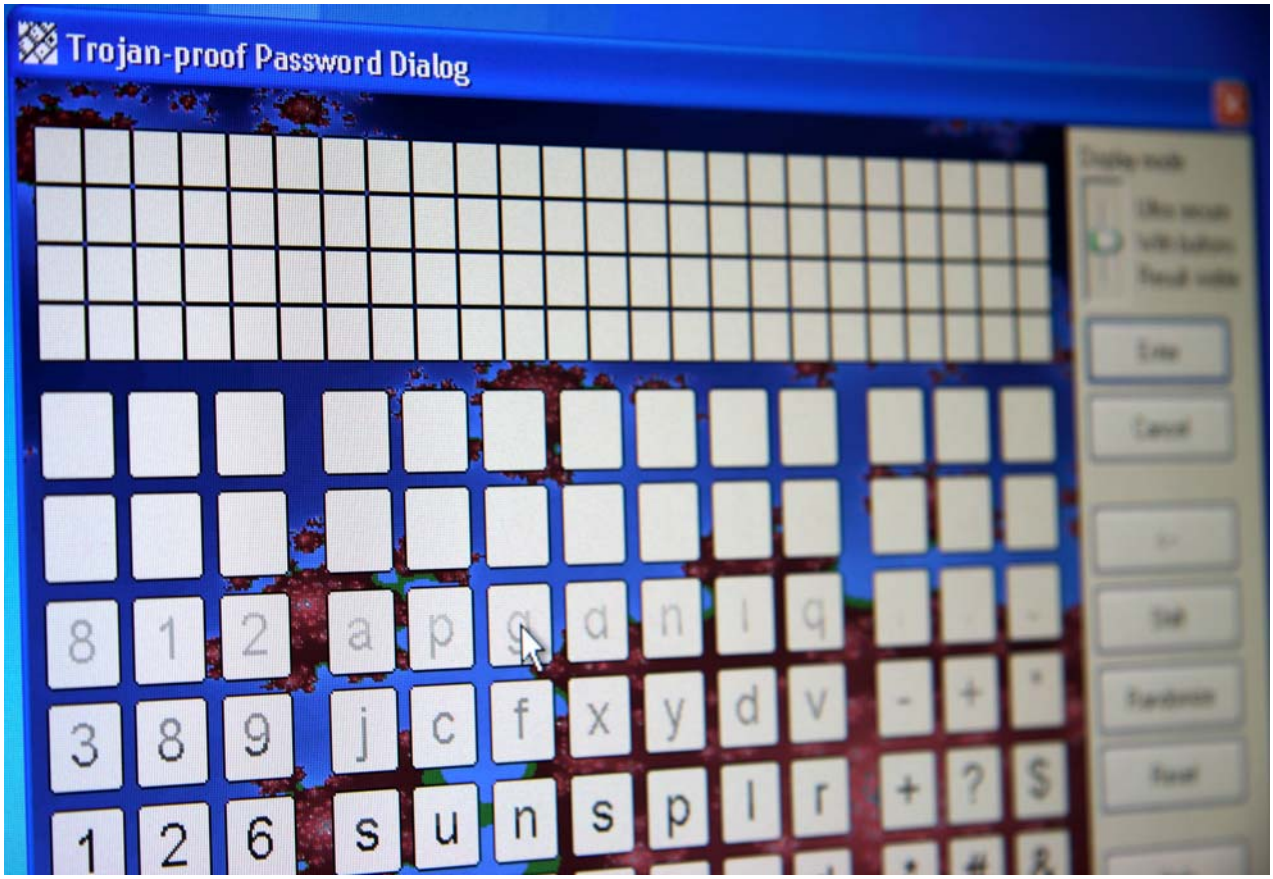



*Back to Index of contents*

# Background information

## Trojan-horse-proof virtual keyboard

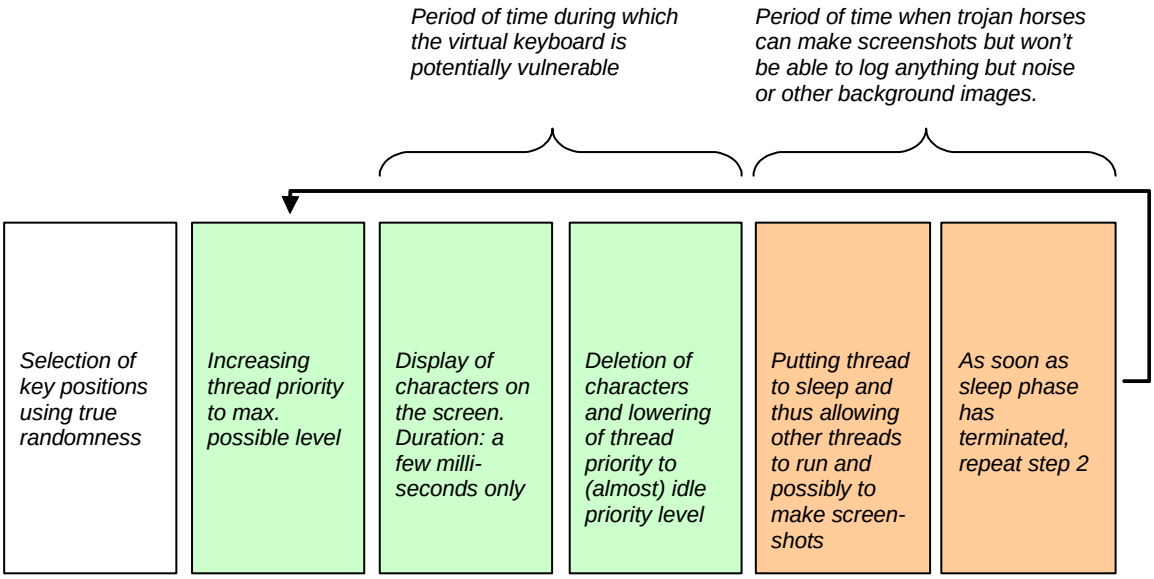
The photo below shows our invention of a virtual keyboard that allows for totally secure password entry. Conventional OTFE (On-The-Fly-Encryption) software or other encryption software only allows for password entry by keyboard (or by smart cards and other authentication methods that are not suitable for guaranteeing perfect secrecy). Keystrokes can either be recorded or transmitted by a hidden piece of hardware in the keyboard or they are simply logged by malicious software – so called “Trojan Horses”. While normal criminals sometimes use the first approach do professional hackers and states generally prefer the latter approach.




In any case does our invention provide perfect secrecy for your passwords as the trojan-horse-proof virtual keyboard does not allow any malicious piece of software to gather any useful information.

The photo actually reveals the mode of operation of the virtual keyboard. Key characters are drawn, deleted, drawn again and deleted again. This sequence repeats itself several times per second. In order to keep highly intelligent trojan horses from taking a snapshot of the screen while all characters are visible, a characteristic that is inherent to the task scheduler, which is implemented in the core of the operating system, is exploited:
Processes running at realtime priority level will probably never be descheduled by processes featuring a lower priority level. Even processes with the same (very high) priority level won’t interrupt our virtual keyboard process as long as the virtual keyboard does not consume too much CPU time. It wouldn’t make must sense for the scheduler to interrupt a media player application by a process that e.g. indexes e-mails in the background. The operating system wouldn’t be popular if audio or video playback was choppy.

Sequence of operations executed by the trojan-horse-proof virtual keyboard:




It should be noted that modern microprocessors feature at least two independent CPU cores. TurboCrypt uses up all available additional CPU cores to compute pseudorandom numbers in order not to give any malicious piece of software any access to CPU time.

This fascinatingly simple but highly efficient method to keep hackers away from your password data has been thoroughly tested many times. You can test the efficiency easily by yourself with the help of a frame grabber tool.

*Back to Index of contents*

## Deniable volumes and deniable encryption

In case you're forced by an adversary to reveal your password, TurboCrypt provides 100% plausible deniability through hidden volumes.

It should be explicitly noted that, although TurboCrypt volumes cannot be identified as a TurboCrypt volume, an adversary can still be sure that you're using encryption software because all encrypted volume files contain "noise". TurboCrypt volumes do not contain any file header or anything else that might identify them as a TurboCrypt volume. Although competitors sometimes pretend that adversaries cannot prove that encryption is used, they can very well do so.
TurboCrypt although can hide volumes in Windows Bitmap files (.BMP) and Audio files (.WAV). TurboCrypt containers (file-hosted volumes) further can have any file extension you like (e.g. .iso, .jpg, .mp3, etc.).

**For extremely tough situations, that is when a TurboCrypt user is forced by somebody else to reveal the password to an encrypted volume, TurboCrypt provides users with the ultimate solution:**

The photo show a real hard disk drive. Enormous amounts of data can be stored on the disk surfaces. Data is stored on individual tracks from the outside to the inside of a disk. Each track is divided into sectors. Sectors are the smallest unit of a disk. A sector is a group of 512 bytes.

The operating system computes for each disk access the sector number and subsequently performs read and write on the selected sector.

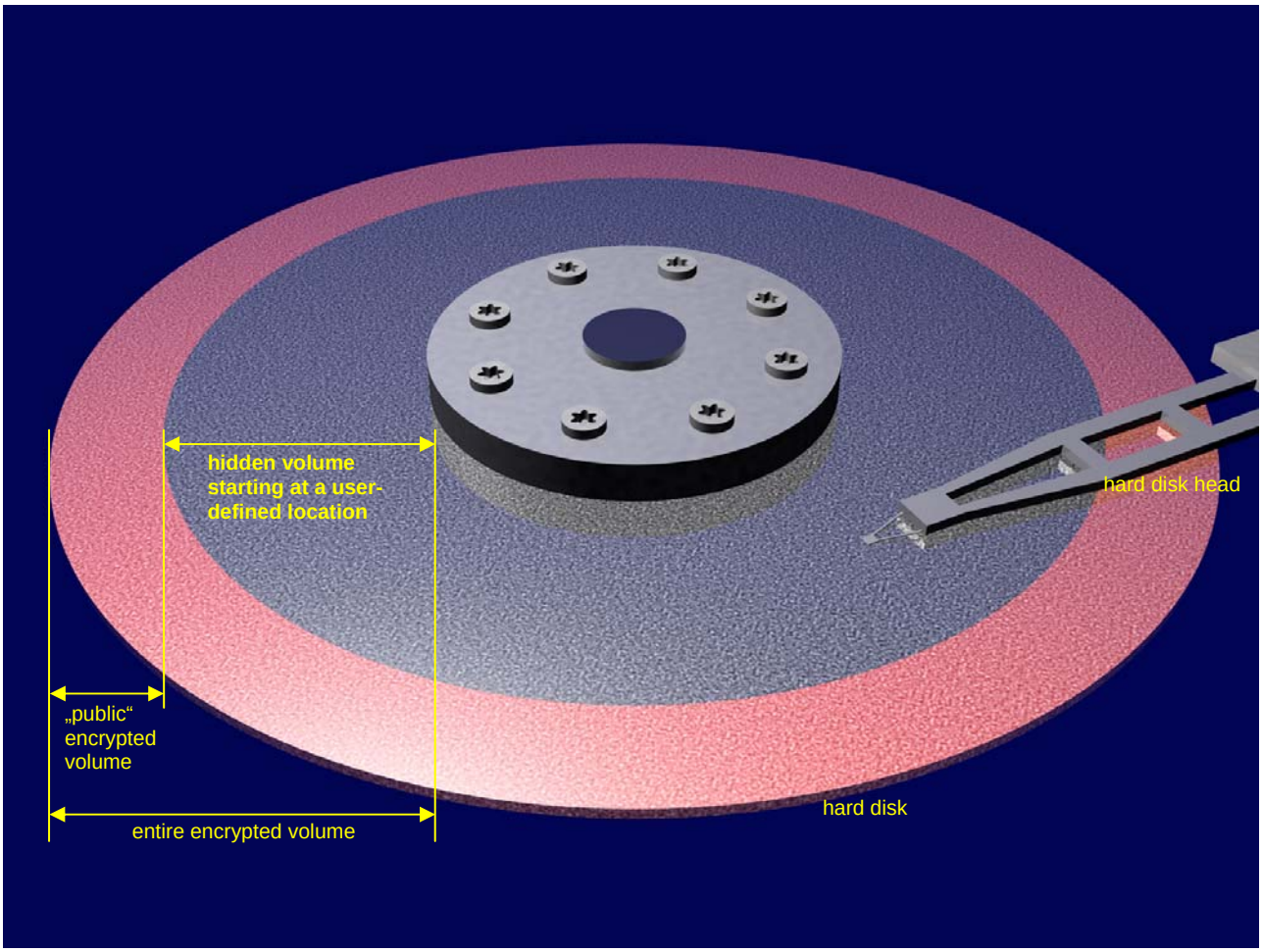
The following picture explains truly deniable encryption.

The hard disk symbolizes an entire volume. The volume is protected with a password that the user can give to anybody who asks for it. In other words, the user will store non-compromising information (e.g. pictures showing himself, Albert Einstein or his wife) there.

Within this "outer" volume is another volume stored. It's a hidden volume (shown in grey/blue color) – one that nobody would expect to find.

Sectors that don't overlap with this new "inner" volume belong to the outer volume only. They are shown in red. As most file systems write information from the start of a disk to the end incrementally, it is possible to occupy unused sectors for other purposes.

It should only be made sure that "unused" sectors of the outer volume don't get suddenly used. In this case would disk space of the outer volume (in red) be insufficient. The file system would simply write to sectors where information of the inner (hidden) volume (grey/blue color) is already stored! **Loss of data in the hidden volume would be the direct result.**
To an attacker the outer volume appears to contain noise. It is impossible for an attacker to identify the sheer existence of an inner and thus highly confidential volume. **During formatting, TurboCrypt writes to all data areas of virtual volumes that could possibly contain a hidden volume, data that looks like noise. Only this ensures TRUE deniability !!!**




TurboCrypt supports (almost) arbitrary start sectors for the inner (hidden) volume!
The lower limit for the start of sector of the hidden volume is the first data sector of the outer volume. If the start of the hidden volume was too close to the start of the outer volume, the outer volume would be corrupted.
The upper limit of the start sector is simply bound to the minimum size of the hidden volume, which is approximately 64Mb.

Users who want to take advantage of the unique feature of TurboCrypt to provide truly deniable hidden volumes must be aware that:

- they MUST remember the start sector they've chosen when they created the volume
- the password of the hidden volume should always be entered using the trojan-horse-proof virtual keyboard
- if they write too much data into the outer volume (which is not deniable and which thus should be secured with a very simple password), the start of the hidden volume CAN EASILY BE OVERWRITTEN !!! Loss of data is the direct result !!!

*Back to Index of contents*

## For more information: http://www.pmc-ciphers.com



Global IP Telecommunications, Ltd. & PMC Ciphers, Inc. - Josephsburgstr. 85, 81673 Munich, Germany
Tel. +49 89 235 1468-0