TrueCrypt is a software application used for on-the-fly encryption (OTFE). It is distributed without cost and the source code is available. It can create a virtual encrypted disk within a file or encrypt a partition or (under MS Windows except Windows 2000) the entire storage device (pre-boot authentication).

Cryptographic algorithms

Individual algorithms supported by TrueCrypt are AES, Serpent and Twofish. Additionally, five different combinations of cascaded algorithms are available: AES-Twofish, AES-Twofish-Serpent, Serpent-AES, Serpent-Twofish-AES and Twofish-Serpent. The cryptographic hash functions used by TrueCrypt are RIPEMD-160, SHA-512 and Whirlpool.

- **Modes of operation**

TrueCrypt currently uses the XTS mode of operation. Prior to this, TrueCrypt used LRW mode in versions 4.1 through 4.3a, and CBC mode in versions 4.0 and earlier. XTS mode is thought to be more secure than LRW mode, which in turn is more secure than CBC mode.
Although new volumes can only be created in XTS mode, TrueCrypt is backward compatible with older volumes using LRW mode and CBC mode. Later versions produce a security warning when mounting CBC mode volumes and recommend that they be replaced with new volumes in XTS mode.

Security concerns

TrueCrypt is vulnerable to various known attacks. To prevent them, the documentation distributed with TrueCrypt requires users to follow various security precautions. Some of those attacks are also detailed below in this section.

- **Plausible deniability**

TrueCrypt supports a concept called plausible deniability, by allowing a single "hidden volume" to be created within another volume. In addition, the Windows versions of TrueCrypt have the ability to create and run a hidden encrypted operating system whose existence may be denied.
The TrueCrypt documentation lists many ways in which TrueCrypt's hidden volume deniability features may be compromised (e.g. by third party software which may leak information through temporary files, thumbnails, etc., to unencrypted disks) and possible ways to avoid this. In a paper published in 2008 and focused on the then latest version (v5.1a) and its plausible deniability, a team of security researchers led by Bruce Schneier states that Windows Vista, Microsoft Word, Google Desktop and others store information on unencrypted disks, which might compromise TrueCrypt's plausible deniability. The study suggested using hidden operating system feature functionality, which was added in TrueCrypt 6.0 and not reviewed (when a hidden operating system is running, TrueCrypt also makes local unencrypted filesystems and non-hidden TrueCrypt volumes read-only to prevent data leaks). The security of TrueCrypt's implementation of this feature was not evaluated because the first version of TrueCrypt with this option had only recently been released.

- **Identifying TrueCrypt volumes**

TrueCrypt volumes do not contain known file headers and their content is indistinguishable from random data, so while it is theoretically impossible to prove that certain files are TrueCrypt volumes, their presence can provide reasonable suspicion (probable cause)[13] that they contain encrypted data. TrueCrypt volume files have file sizes that are evenly divisible by 512 and their content passes chi-square randomness tests. These features give reason to suspect a file to be a TrueCrypt volume[14].

If system drive or a partition on it has been encrypted then the above paragraph applies to the contents of that drive/partition. However the boot loader is replaced with a TrueCrypt one to allow a password to be entered and the decryption to begin. By default this clearly states that it is the TrueCrypt boot loader when run. It may be customized to display a BIOS-like message (such as "operating system missing"), although this reduces the functionality of the boot loader and still results in a flicker of the hard-drive light when return is pressed (as the entered password is checked). In either case, offline analysis of the drive can be used to determine that a TrueCrypt boot loader is present.

- **Passwords stored in memory**

TrueCrypt stores its keys in RAM; on an ordinary personal computer the DRAM will maintain its contents for several seconds after power is cut (or longer if the temperature is lowered). Even if there is some degradation in the memory contents, various algorithms can intelligently recover the keys. This method (which would apply in particular to a notebook computer stolen while in power-on, suspended, or screen-locked mode) has been successfully used to attack a file system protected by TrueCrypt.

- **Physical security**

TrueCrypt documentation states that TrueCrypt is unable to secure data on a computer if an attacker physically accessed it and TrueCrypt is used on the compromised computer by the user again (this does not apply to a common case of a stolen or lost computer).[16] The attacker having physical access to a computer can, for example, install a hardware/software keylogger, a bus-mastering device capturing memory, or install any other malicious hardware or software, allowing the attacker to capture unencrypted data (including encryption keys and passwords), or to decrypt encrypted data using captured passwords or encryption keys. Therefore, physical security is a basic premise of a secure system.

- **The "Stoned" bootkit**

The "Stoned" bootkit, an MBR rootkit presented by Austrian software developer Peter Kleissner at the Black Hat Technical Security Conference USA 2009, has been shown capable of tampering TrueCrypt's MBR effectively bypassing TrueCrypt's full volume encryption.[19][20] (but potentially every hard disk encryption software is affected too if it does not rely on hardware-based encryption technologies like TPM, or—even if it does—if this type of attack is made with administrative privileges while the encrypted operating system is running).
Two types of attack scenarios exist in which it is possible to maliciously take advantage of this bootkit, currently written for Win32 platforms only: in the first one, the user is required to launch the bootkit with administrative privileges once the PC has already booted into Windows; in the second one, analogously to hardware keyloggers, a malicious person needs physical access to the user's TrueCrypt-encrypted hard disk: in this context this is needed to modify the user's TrueCrypt MBR with the Stoned's one and then place the hard disk back on the unknowing user's PC, so that when the user boots the PC and types his/her TrueCrypt password on boot, the "Stoned" bootkit intercepts it thereafter because, from that moment on, the Stoned bootkit is loaded before TrueCrypt's MBR in the boot sequence. The first type of attack can be prevented as usual by good security practices, i.e. avoid running non-trusted executables with administrative privileges. The second one can be successfully neutralized, by the user if

he/she suspects that the encrypted hard disk might have been physically available to someone he/she doesn't trust, by booting the encrypted operating system with TrueCrypt's Rescue Disk instead of booting it directly from the hard disk and restoring boot loader in MBR.