## Encrypting files

All modern-day operating systems will have some support for encrypting files. You can also use utilities, such as WinZip to encrypt your files.

### Windows

In Microsoft Windows.

1. A right click on a file or folder and select Properties.
2. Select the Advanced button and select the Encrypt contents to secure data check box.
3. Select OK to close the Advanced Attributes window, select Apply, and then select OK.

To encrypt a folder.

1. Create a folder you would like to encrypt.
2. Go to Disk Utility.
3. Click File > New Image > Image from Folder.
4. Find and click on the folder you created.
5. Select an encryption method, create a password, and save your encrypted folder.

### Linux

To encrypt in a Linux-based operating system, such as Ubuntu.

1. Right-click on a file or folder and select Encrypt.
2. Select to use a shared passphrase or create or import a key.
3. Once you have gone through the process of a passphrase or key, and if you are encrypting a folder, you will be asked if you want to encrypt each file contained within or pack them together in a compressed file.

**Video**

The videos below give detailed guides to encryption for the main operating systems from TechJunkie, Techmyoldtechchannel and TechJuMax Dalton.

ICTICT532 - Encryption files[67]

[67] https://www.youtube.com/embed/videoseries?list=PLr24-CFnxcCyDFwQnWiiqVKGn1SmLpgcC

## Encrypting hard drives

### Windows device encryption

Turn on device encryption

1. Sign into Windows with an administrator account (you may have to sign out and back in to switch accounts). For more info, see Create a local or administrator account in Windows 10.
2. Select the **Start** button, then select **Settings,** then select **Update & Security** , select **Device encryption**.

If **Device encryption** does not appear, it is not available. You may be able to use standard BitLocker encryption instead.

Open Device encryption setting.

1. If device encryption is turned off, select **Turn on**.

### Windows BitLocker encryption

Turn on standard BitLocker encryption

1. Sign into your Windows device with an administrator account (you may have to sign out and back in to switch accounts). For more info, see Create a local or administrator account in Windows 10.
2. In the search box on the taskbar, type **Manage BitLocker** and then select it from the list of results. Or, you can select the **Start** button, and then under **Windows System** , select **Control Panel**. In **Control Panel** , select **System and Security** , and then under **BitLocker Drive Encryption** , select **Manage BitLocker**.
3. Note: You will only see this option if BitLocker is available for your device. It is not available on Windows 10 Home edition.
4. Select **Turn on BitLocker** and then follow the instructions.

### Mac OS X Lion or later

When FileVault is turned on, your Mac always requires that you log in with your account password.

1. Choose **Apple menu** and select **System Preferences** , then select **Security & Privacy**.
2. Click the **FileVault** tab.
3. Click **Locked** , then enter an administrator name and password.
4. Click **Turn On FileVault**.

### Linux OS

Most Linux operating systems do not allow for an easy encryption of a hard drive after the initial installation process is complete. The option to encrypt is selected during the steps of the operating system install.

There are optional workarounds; however, it is considered easier to back up your data and re-install the operating system from scratch.

**Video**

Watch the first video by Britec09 to learn how to encrypt a Windows 10 hard drive and the second, by Hands-On Mac on how to use Filevault for a Mac.

The videos will play one after the other, but if you want to watch them at separate times use the playlist menu in the top right corner of the video.

ICTICT532 - Encrypting hard drives[68]

[68] https://www.youtube.com/embed/videoseries?list=PLr24-CFnxcCy39fyH-3HlmUSlvk661Fyf

## Hashing

Hashing is used to verify data integrity; it is not encryption. Hashing is achieved by having an algorithm performed on data, such as a file or message, to create a number called a hash (also known as a checksum). The hash is then used to validate that the data has not been modified, tampered with, or corrupted. This is where you can verify the data has maintained integrity.

Integrity of the data is directly linked to the hashing algorithm, if the hash differs at any point, it means that the data has been modified or corrupted.

### Example

An IT technician has to update the BIOS on a computer system. When running the actual update to the ROM (Read Only Memory) it applies the data in sections and compares the section that was just updated with the checksum, which is the hash. If the checksum differs, it then reapplies the section again just in case it was a packet of data that did not transfer over properly.

If the checksum fails again, the whole process is stopped, and the BIOS returned to its original condition.

### Video

Watch the videos from Lisk and Programming with Professor Sluiter, for more information about hashing and encryption.

The videos will play one after the other, but if you want to watch them at separate times use the playlist menu in the top right corner of the video.

ICTICT532 - Hashing[69]


203379583 / Gustavo Frazao / shutterstock.com

[69] https://www.youtube.com/embed/videoseries?list=PLr24-CFnxcCzYWCJVYjGEbjbH2B10UtYk

## Hashing algorithms

Some of the common hashing algorithms in use today are detailed below.

### MD5

Message Digest 5 (MD5) is a common hashing algorithm that produces a 128-bithash. Hashes are commonly shown in hexadecimal format instead of a stream of 1s and 0s. For example, an MD5 hash is displayed as 32 hexadecimal characters instead of 128 bits. Hexadecimal characters are composed of 4 bits and use the numbers 0 through 9 and the characters a through f.

### SHA

Secure Hash Algorithm (SHA) is another hashing algorithm. There are several variations of SHA grouped into four families—SHA-0, SHA-1, SHA-2, and SHA-3:

- SHA-0 is not used.
- SHA-1 is an updated version that creates 160-bit hashes. This is similar to the MD5 hash except that it creates 160-bit hashes instead of 128-bit hashes.
- SHA-2 improved SHA-1 to overcome potential weaknesses. It includes four versions. SHA-256 creates 256-bit hashes and SHA-512 creates 512-bit hashes. SHA-224 (224-bit hashes) and SHA-384 (384-bit hashes) create truncated versions of SHA-256 and SHA- 512, respectively.
- SHA-3 (previously known as Keccak) is an alternative to SHA-2. The U.S. National Security Agency (NSA) created SHA-1 and SHA-2. SHA-3 was created outside of the NSA and was selected in a non-NSA public competition. It can create hashes of the same size as SHA-2 (224 bits, 256 bits, 384 bits, and 512 bits).

### HMAC

Another method used to provide integrity is with a Hash-based Message Authentication Code (HMAC). An HMAC is a fixed-length string of bits similar to other hashing algorithms such as MD5 and SHA-1 (known as HMAC-MD5 and HMAC-SHA1). However, HMAC also uses a shared secret key to add some randomness to the result and, only the sender and receiver know the secret key.

### RIPEMD

RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is another hash function used for integrity. It is not as widely used as MD5, SHA, and HMAC.

## Hashing files and messages

Numerous applications calculate and compare hashes automatically, this normally occurs without any user involvement. An example is where a digital signature uses a hash within an email, and email applications automatically create and compare the hashes.

Additionally, there are a number of applications you can use to manually calculate hashes. As an example, the Microsoft Store has a tool developed by DigitalVolcano Software called **Hash Tool**, which is a free program anyone can use to create hashes of files.

### Weblink

Visit the link below to see the Digital Volcano hash tool.

Hash Tool[70]

A Google search on “hash tool download” will show a list of applications that are available.

### Note

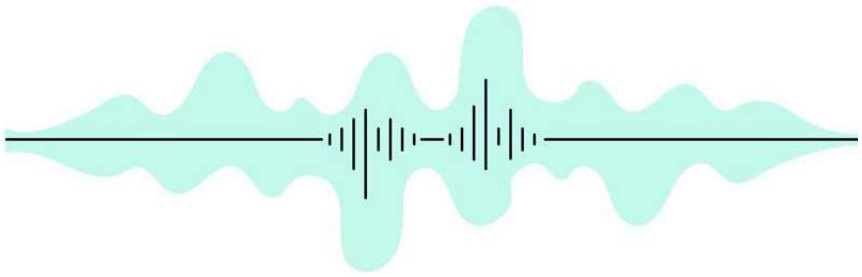
It is worth emphasising that hashes are one-way function. You can calculate a hash on a file or a message, but you cannot use the hash to reproduce the original information. Also, to note, the hash code will always be the same length, no matter how big the file is.

### Research

Download a hashing algorithms application to your system.

Copy a number of files to a temporary folder and create a hash for the file.

Compare the hash and see if there is any correlation between each created hash.



1381983182 / Iriejah / shutterstock.com

[70] https://www.digitalvolcano.co.uk/hash.html

### Hashing messages

For integrity when sending for messages, hashing is a good option. Hashing will provide guarantee to a person receiving a message that the message has not been altered or modified.

**Scenario**

If we look at a scenario where Bill is sending a message to Ted. The message is “The song is almost complete.” As this message may not be secret, there is no need to encrypt it. Although, we do want to establish integrity, so the focus on this description is only on hashing.

In this scenario, something modified the message before it reaches Ted. When Ted receives the message and the original hash, the message is now “The song is nearly almost complete.”

It shows that the message has been modified in transit, but the hash has not changed. A program on Ted’s computer calculates the MD5 hash on the received message as 564294439E1617F5628A3E3EB75643FE. It then compares the received hash with the calculated hash.

- Hash created on Bill’s computer, and received by Ted’s computer: D9B93C99B62646ABD06C887039053F56
- Hash created on Ted’s computer: 564294439E1617F5628A3E3EB75643FE

The above shows that the hashes are different, so it indicates that the message has lost integrity. The program on Ted’s computer would report the discrepancy.

**Note**

This shows that there could have been a malicious attacker modifying the message, or it could have been a technical problem. However, Ted is now aware that the received message is not the same as the sent message and he should not trust it.