Encrypting a single file

John C. Nash

2024-04-19

Abstract

This document gives some suggestions on how to encrypt, decrypt and view a single (text) file in different computing environments.

Introduction

Encryption is a key aspect of modern computing and communications. In this article, we will focus on the encryption of a single **file** of data, normally assumed to be ordinary text, as well as its decryption and viewing. We will usually restrict our attention to **symmetric** encryption, where the same **password** or **passphrase** is used to encrypt and decrypt the file. However, there are a number of important tools that use **passkeys** and may use asymmetric or public-private key pairs.

Note that encryption can be applied to whole storage devices, volumes, directories or streams. Here we will not explore those possibilities, even if our tools could be used for such applications.

Required tools

To simplify our discussion, we will give our tools the *generic* names

- **ecrypt**, a program that encrypts a file from FILE to FILE.CCC
- **dcrypt**, a program that decrypts a file from FILE.CCC to FILE
- **vcrypt**, a program that displays FILE.CCC or part thereof for human reading, hopefully removing all readable traces of the decrypted information as it exits. This facility is not available for all encryption tools, so the user may have to take measures to avoid leaving unencrypted information accessible to those not authorized to view it.

Some possible tools

ccrypt

ccrypt (http://ccrypt.sourceforge.net/) is intended for file encryption. I happen to be acquainted with its author, Peter Seligman of Dalhousie University. It is available for most platforms.

- ecrypt becomes

  ccrypt FILE

  and outputs FILE.cpt

  The password must be entered twice, though there are options to include it on the command line, but such options are discouraged for reasons of security.

- dcrypt becomes (the user is prompted for the password)

  ccrypt -d FILE.cpt

- vcrypt becomes

  ccat FILE.cpt

All the above are executed in a terminal (or command) window on Linux, Windows and (we presume) Macintosh. Note the **qccrypt** GUI wrapper for Linux and Windows.

Linux

ecrypt-ccrypt.sh

```
#!/bin/bash
#
# ecrypt-ccrypt.sh -- use ccrypt to encrypt file that is parameter 1
# For Linux
#
# J C Nash 2024-3-28
# Uses Peter Seligman ccrypt as encryption engine
#
#
ccrypt $1
```

dcrypt-ccrypt.sh

```
#!/bin/bash
#
# dcrypt-ccrypt.sh -- use ccrypt to decrypt file that is parameter 1
# For Linux
#
# J C Nash 2024-3-28
# Uses Peter Seligman ccrypt as encryption engine
#
#
ccrypt -d $1
```

vcrypt-ccrypt.sh

```
#  vcrypt-ccrypt.bat -- use ccrypt to view encrypted file that is parameter 1
#  J C Nash 2024-3-28
#  For Linux
#
#  Uses Peter Seligman ccrypt as encryption engine
#
#  Could use:
#  ccat %1
#  but that does not paginate
stty -echo
read -p "Password: " passw; echo
stty echo
ccat -K $passw  $1 | less
# #  safety!
unset passw
echo "Check no password:"
```

```
echo $passw
echo "That's all"
```

There are variants of the GUI wrapper **qccrypt** for several Linux distros as well as source code at http://qccrypt.free.fr/download.html .

Windows

Note that these tools are intended to be run inside a command terminal screen. However, there are 32-bit installable and portable Windows packages at, respectively, http://qccrypt.free.fr/download/qccrypt-0.9.1.msi and http://qccrypt.free.fr/download/qccrypt-0.9.1-win32.zip .

ecrypt-ccrypt.bat

```
REM ecrypt-ccrypt.bat -- use ccrypt to encrypt file that is parameter 1
REM For Windows
REM J C Nash 2024-3-28
REM
REM
REM Uses Peter Seligman ccrypt as encryption engine
REM
REM
ccrypt %1
```

dcrypt-ccrypt.bat

```
REM dcrypt-ccrypt.bat -- use ccrypt to decrypt file that is parameter 1
REM For Windows
REM J C Nash 2024-3-28
REM
REM
REM Uses Peter Seligman ccrypt as encryption engine
REM
REM
ccrypt -d %1
```

vcrypt-ccrypt.bat

```
REM   vcrypt-ccrypt.bat -- use ccrypt to view encrypted file that is parameter 1
REM   J C Nash 2024-3-28
REM   For Linux
REM
REM   Uses Peter Seligman ccrypt as encryption engine
REM
REM   Could use:
REM   ccat %1
REM   but that does not paginate
?? REM  Do the following ideas work in Windows??
REM  stty -echo
REM  read -p "Password: " passw; echo
REM  stty echo
REM  ccat -K $passw  $1 | less
REM  REM   safety!
REM  unset passw
REM  echo "Check no password:"
REM  echo $passw
REM  echo "That's all"
```

```
set /p PWD="Password: "
cls
REM   echo %PWD%    REM test only
ccrypt -c -K %PWD% %1 | more
```

Android

There is a tarball of an Android program

https://ccrypt.sourceforge.net/download/1.10/ccrypt-1.10.android.tar.gz

but I have NOT determined how to install and use this.

7zip

7zip (https://7-zip.org/) is free and open source software intended primarily for compressed archiving of collections of files. However, it includes strong AES-256 encryption in 7z and ZIP formats. Moreover, it is available for almost all platforms, including Android. However, vcrypt seems unavailable in a form that is "view only".

Linux

ecrypt-7z.sh

```
#!/bin/bash
#
# ecrypt-7z.sh -- use 7z to encrypt file that is parameter 1
# For Linux
#
# J C Nash 2024-4-10
# Uses 7z from package 7zip as encryption engine
#
7z a -p "$1.zip" "$1"
echo "That's all"
```

dcrypt-7z.sh

```
#!/bin/bash
#
# dcrypt-7z.sh -- use 7z to decrypt file that is parameter 1
# For Linux
#
# J C Nash 2024-3-28
# Uses 7z from 7zip package as encryption engine
#
#
7z e $1
```

Windows

ecrypt-7z.bat

```
REM ecrypt-7z.bat -- use 7z to encrypt file that is parameter 1
REM  For Windows
REM
REM  J C Nash 2024-4-18
REM  Uses 7z from 7z2404-x64.exe (7zip encryption engine)
```

```
REM
7z a -p "%1.zip" "%1"
echo "That's all"
```

dcrypt-7z.bat

```
REM !/bin/bash
REM
REM  dcrypt-7z.BAT -- use 7z to decrypt file that is parameter 1
REM  For Windows
REM
REM  J C Nash 2024-3-28
REM  Uses 7z from 7z2404-x64.exe as encryption engine
REM
REM should figure out how to rename file if there is a name collision
7z e "%1"
```

Mac ??

Android

I have found two "apps" on Android that I have verified can unpack password protected zip files using the encryption of the 7z family of software: **7zipper** and **Zarchive**. I will NOT provide programs to encrypt and decrypt for Android, as "Android" is actually several different operating systems in my opinion, and I find the various "help" postings on the Internet and in Youtube never fit my own devices (two tablets and two phones, all slightly different). A particular annoyance is that Zarchiver does not allow decryption of files on a plug-in SD memory card for some reason. There are a number of postings and videos on how to "fix" this, but I did not find any worked. If I copied a zip archive to the Internal memory device of my tablet, then Zarchiver would decrypt it so a text editor (I used the Jota Text Editor) could view the file.

Two particular concerns with mobile phones and tablets is that the decrypted (i.e., plaintext) document is written to the storage of the device. This means

- for security, it must be properly erased. This means over-writing the storage after the control entry has been modified, as the actual data will still be present otherwise;
- with the almost certain network connectivity, it is difficult to be sure that external agents cannot view the plaintext material before we can erase it. I do not have the expertise to estimate this risk, but it is certainly not zero.

gpg

- ecrypt becomes

  gpg –symmetric FILE

  and outputs FILE.gpg

  The password must be entered twice, though there are options to include it on the command line, but such options are discouraged for reasons of security.

- dcrypt becomes (the user is prompted for the password)

  gpg –decrypt FILE.gpg

- There does not appear to be a reasonable option with gpg for vcrypt

Linux

ecrypt-gpg.sh

```
#!/bin/bash
#
# ecrypt-gpg.sh -- use gpg to encrypt file that is parameter 1
# For Linux
#
# J C Nash 2024-4-10
# Uses gpg (gnupg) as encryption engine
#
stty -echo
read -p "Password: " passw; echo
stty echo
gpg --batch -o "$1.gpg" --symmetric --passphrase $passw "$1"
# #  safety!
unset passw
echo "Check no password left around:"
echo $passw
echo "That's all"
```

dcrypt-gpg.sh

```
#!/bin/bash
#
# dcrypt-gpg.sh -- use gpg to decrypt file that is parameter 1
# For Linux
#
# J C Nash 2024-4-10
# Uses gpg (gnupg) as encryption engine
#
stty -echo
read -p "Password: " passw; echo
stty echo
gpg -d -o "$1.txt" --passphrase $passw "$1"
# #  safety!
unset passw
echo "Check no password left around:"
echo $passw
echo "That's all"
```

Windows

Note that these tools are intended to be run inside a command terminal screen.

ecrypt-gpg.bat

```
REM ecrypt-gpg.bat -- use gpg to encrypt file that is parameter 1
REM For Windows
REM J C Nash 2024-3-28
REM
REM
REM
REM Not yet developed
REM
gpg -r --encrypt --symmetric %1
```

dcrypt-gpg.bat

```
REM dcrypt-gpg.bat -- use gpg to decrypt file that is parameter 1
REM For Windows
REM J C Nash
REM
REM
REM
REM Not yet developed
REM
?? gpg ??  %1
```

Strategies for use

The fact that different platforms have generally different modes of use forces consideration of which tools to use for file encryption in a particular situation. Some of the questions a user must answer are as follows:

- Will files encrypted by a particular tool require decryption on a different platform. This is particularly important, for example, in the case of ccrypt, since it is not reasonably available on Android (and we do not know about iPhone)