???????-Encrypt

??????

Linux

1. Cryptsetup - ?? LUKS/plain dm-crypt/loop-AES/TrueCrypt/BitLocker ?????
   - How to Encrypt and Decrypt a Partition in Linux
2. eCryptFS
3. EncFS
4. CryFS
5. VeraCrypt - ???????????????
   - How to Encrypt USB Drive on Linux Using VeraCrypt

Windows

1. AxCrypt

????

??

- EncryptFS ????? CentOS 6.x
- EncFS ????? CentOS 6???? EPEL ?? fuse-encfs
- CryFS ??? Ubuntu

????

`<encrypt-command> ~/ENCRYPTED-DIR ~/DECRYPTED-DIR`

- ??
- ??
  ??
- ??
- ??
- ??????????????????????

????

1. ?????????? Dropbox ??????????? Dropbox ????????????????????????????????? dropbox ????????????????????? Dropbox ?????????????????
2. ????????????????????? NFS ???NAS?????? Linux ????????????? NAS ??

??????

??
???????

GPG

????????????

```
[root@db2v9 gpg]# tree
.
├── npp.7.8.8.Installer.exe
├── npp.7.8.8.Installer.exe.sig
└── nppGpgPub.asc

0 directories, 3 files
[root@db2v9 gpg]#
[root@db2v9 gpg]# gpg --verify npp.7.8.8.Installer.exe.sig npp.7.8.8.Installer.exe
gpg: Signature made Sun 28 Jun 2020 11:02:09 PM CST using RSA key ID 8D84F46E
gpg: Can't check signature: No public key

[root@db2v9 gpg]# gpg --import nppGpgPub.asc
gpg: key 8D84F46E: public key "Notepad++ <don.h@free.fr>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
gpg: no ultimately trusted keys found

[root@db2v9 gpg]# gpg --verify npp.7.8.8.Installer.exe.sig npp.7.8.8.Installer.exe
gpg: Signature made Sun 28 Jun 2020 11:02:09 PM CST using RSA key ID 8D84F46E
gpg: Good signature from "Notepad++ <don.h@free.fr>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 14BC E436 2749 B2B5 1F8C  7122 6C42 9F1D 8D84 F46E
```

????

- GPG ?????
  - Linux: HowTo Encrypt And Decrypt Files With A Password
  - How to easily encrypt a file with GPG on Linux

    - Encrypting and decrypting files with password in Linux
- ????
    - Linux: HowTo Encrypt And Decrypt Files With A Password
    - 7 Tools to Encrypt/Decrypt and Password Protect Files in Linux
- LUKS
    - Implementing corporate laptop encryption using LUKS
    - How to unlock LUKS using Dropbear SSH keys remotely in Linux - nixCraft (cyberciti.biz)

??????? - Cryptsetup

??

??????? USB??????????????????????????

????

????

```
sudo apt install cryptsetup
```

????????????????? /dev/sdb??? GParted ????????? /dev/sdb1???? ext4 ??

```
sudo cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb1
```

> ??????????? ?? YES?
> ??????????

???????????

- ?????????????????
- ????????????????????? `/dev/mapper/<input-name>`

```
sudo cryptsetup luksOpen /dev/sdb1 sdb1
sudo fdisk -l
sudo mkfs.ext4 /dev/mapper/sdb1
sudo tune2fs -m 0 /dev/mapper/sdb1
```

???????????

```
sudo e2label /dev/mapper/sdb1 MyCCWallet
```

??????????????????????????

```
sudo mkdir /mnt/encrypted
sudo mount /dev/mapper/sdb1 /mnt/encrypted
sudo touch /mnt/encrypted/test.txt
```

??????????????????

```
sudo umount /dev/mapper/sdb1
sudo cryptsetup luksClose sdb1
```

Linux Mint ????

?????????????? Linux Mint ??

ask_password.png

Optional: ????

How to find and verify which Luks slot a passphrase is in on Linux

```
# Do not activate device, just check
sudo cryptsetup --verbose open --test-passphrase /dev/vda3

Enter passphrase for /dev/vda3:
Key slot 0 unlocked.
Command successful.
```

Learning Cryptsetup

- How to change LUKS disk encryption passphrase in Linux
- Create an encrypted file vault on Linux

Learning

GPG

- Using GPG to Encrypt and Decrypt Files on Linux [Hands-on for Beginners]

LUKS

- How to unlock LUKS using Dropbear SSH keys remotely in Linux - nixCraft (cyberciti.biz)
- Linux Hard Disk Encryption With LUKS [cryptsetup command] - nixCraft (cyberciti.biz)

OpenSSL

Create Example Reference File, let us create a 1GB large text file using the fallocate command:

```
fallocate -l 1024M test.txt
echo "LinuxShellTips tutorial on encrypting a large file with OpenSSL in Linux" >> test.txt
cat test.txt
```

Encrypt File with Password (?????)

```
openssl enc -aes-256-cbc -pbkdf2 -p -in test.txt -out test.txt.enc
```

- enc executes the symmetric key encryption process.
- -aes-256-cbc specifies the use of 256 bits cryptographic key.
- -pbkdf2 is the default algorithm being used.
- -p prints used salt, key, and IV.
- -in points to the input file.
- -out points to the output file.

To decrypt the file, run:

```
openssl aes-256-cbc -d -pbkdf2 -in test.txt.enc -out sample_decrypted.txt
```

> ❝ You will be required to enter the encryption password you generated earlier.

Encrypt File with Key (?????)

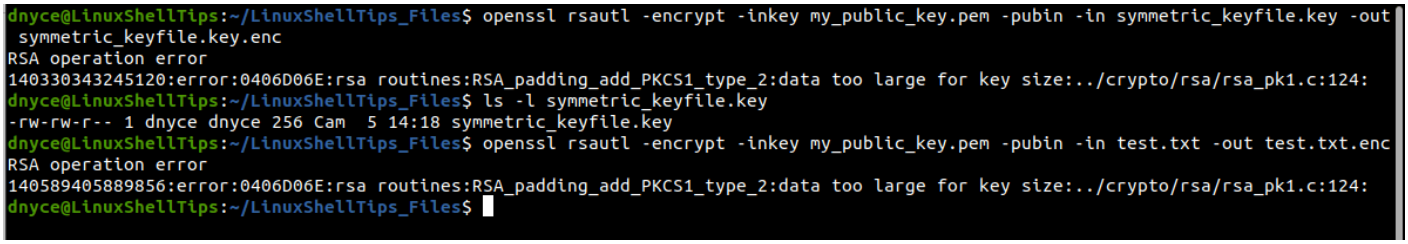
```
# generate a key file
openssl rand 256 > symmetric_keyfile.key
# use the keyfile to encrypt our file
openssl enc -in test.txt -out test.txt.enc -e -aes-256-cbc -pbkdf2 -k symmetric_keyfile.key
```

To decrypt the file, run:

```
openssl enc -in test.txt.enc -out draft_decrypted.txt -d -aes-256-cbc -pbkdf2 -k
symmetric_keyfile.key
```

?????? (Asymmetric Encryption)

??????????????????????????: data too large for key size.



> TIP:
> ??
> ???

Hashing

????

```
# For file
openssl dgst -sha256 my.file

# For string
echo -n "HelloWorld" | openssl dgst -sha256
```

Generate a token key

```
openssl rand -hex 32
```

????

- OpenSSL ??????????????????