

```

from cryptography.fernet import Fernet

#
# encrypt
#
key = Fernet.generate_key()      # save up this key

in_file_name = 'original.txt'    # plain text
out_file_name = 'encrypted-original.txt' # encrypted file
with open(in_file_name, "rb") as fin, open(out_file_name, "wb") as fout:
    while True:
        block = fin.read(65536)
        if not block:
            break
        f = Fernet(key)
        output = f.encrypt(block)
        fout.write(output)

#
# decrypt
#
in_file_name = 'encrypted-original.txt' # encrypted file
out_file_name = 'original2.txt'        # plain text again
with open(in_file_name, "rb") as fin, open(out_file_name, "wb") as fout:
    while True:
        block = fin.read(87480)
        if not block:
            break
        f = Fernet(key)
        output = f.decrypt(block)
        fout.write(output)

```

NOTE: For each block of 65536 bytes plain text, Fernet generates 87480 bytes of encrypted data. One can use 1024:::1464 apart from 65536:::87480 combination. Both of these ratios have been successful in handling files > 30GB.