Decryption Utilizing Linux Bash Commands

```
analyst@f295245bb0d9:~$ ls
file1.txt  file2.txt
analyst@f295245bb0d9:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .profile  file1.txt  file2.txt
analyst@f295245bb0d9:~$ cat file1.txt
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
analyst@f295245bb0d9:~$ cat file2.txt
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
9sxa5Yq20Ranalyst@f295245bb0d9:~$ sha256sum file1.txt
131f95c51cc819465fa1797f6ccacf9d494aaaff46fa3eac73ae63ffbdfd8267  file1.txt
analyst@f295245bb0d9:~$ sha256sum file2.txt
2558ba9a4cad1e69804ce03aa2a029526179a91a5e38cb723320e83af9ca017b  file2.txt
analyst@f295245bb0d9:~$ sha256sum file1.txt >> file1hash
analyst@f295245bb0d9:~$ sha256sum file2.txt >> file2hash
analyst@f295245bb0d9:~$ cmp file1hash file2hash
file1hash file2hash differ: char 1, line 1
analyst@f295245bb0d9:~$
```

```
analyst@70149bbaf268:~$ ls
Q1.encrypted  README.txt  caesar
analyst@70149bbaf268:~$ cat README.txt
Hello,
All of your data has been encrypted. To recover your data, you will need to solve a cipher. To get started look for a hidden file in the caesar subdirectory.
analyst@70149bbaf268:~$ ls -la
total 40
drwxr-xr-x 3 analyst analyst 4096 Jun  8 19:09 .
drwxr-xr-x 1 root    root    4096 Jun  8 18:02 ..
-rw------- 1 analyst analyst   24 Jun  8 19:10 .bash_history
-rw-r--r-- 1 analyst analyst  220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 analyst analyst 3574 Jun  8 18:02 .bashrc
-rw-r--r-- 1 analyst analyst 3574 Jun  8 18:02 .profile
-rw-r--r-- 1 root    root     260 Jun  8 18:02 Q1.encrypted
-rw-r--r-- 1 root    root     165 Jun  8 18:02 README.txt
drwxr-xr-x 2 root    root    4096 Jun  8 18:02 caesar
analyst@70149bbaf268:~$ cd caesar
analyst@70149bbaf268:~/caesar$ ls
analyst@70149bbaf268:~/caesar$ ls -a
.  ..  .leftShift3
analyst@70149bbaf268:~/caesar$ cat .leftShift3
Lq rughu wr uhfryhu brxu ilohv brx zloo qhhg wr hqwhu wkh iroorzlqj frppdqg:

rshqvvo dhv-256-fef -sengi2 -d -g -lq T1.hqfubswhg -rxw T1.uhfryhuhg -n hwwxeuxwh
analyst@70149bbaf268:~/caesar$ cat .leftShift3 | tr "d-za-cD-ZA-C" "a-zA-Z"
In order to recover your files you will need to enter the following command:

openssl aes-256-cbc -pbkdf2 -a -d -in Q1.encrypted -out Q1.recovered -k ettubrute
analyst@70149bbaf268:~/caesar$ ^C
analyst@70149bbaf268:~/caesar$ openssl aes-256-cbc -pbkdf2 -a -d -in Q1.encrypted -out Q1.recovered -k ettubrute
Can't open Q1.encrypted for reading, No such file or directory
139921270850752:error:02001002:system library:fopen:No such file or directory:../crypto/bio/bss_file.c:69:fopen('Q1.encrypted','r')
139921270850752:error:2006D080:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:76:
analyst@70149bbaf268:~/caesar$ cd ~
analyst@70149bbaf268:~$ openssl aes-256-cbc -pbkdf2 -a -d -in Q1.encrypted -out Q1.recovered -k ettubrute
analyst@70149bbaf268:~$ ls
Q1.encrypted  Q1.recovered  README.txt  caesar
analyst@70149bbaf268:~$ cat Q1.recovered
If you are able to read this, then you have successfully decrypted the classic cipher text. You recovered the encryption key that was used to encrypt this file.
analyst@70149bbaf268:~$
```

Some of the file decryption process with Linux Bash CLI

During the activity, I performed the following steps:

- I started by listing the contents of the current directory using the `ls` command and confirmed the presence of two files, `file1.txt` and `file2.txt`.
- I used the `cat` command to display the contents of both files and observed that the contents appeared identical.
- To determine if the files were actually different, I generated the hash values for each file using the `sha256sum` command.
- I compared the generated hash values and found that they were different, indicating that the files were indeed different.

- Next, I wrote the hash values to separate files using the `sha256sum` command with the output redirection (`>>`) operator.
- To confirm the differences in the hash values, I used the `cat` command to display the hash values in the respective files.
- Finally, I used the `cmp` command to compare the hash values and identified the first difference at the first character of the first line.

Throughout this process, I learned the following:

- The `sha256sum` command is used to generate hash values for files using the SHA-256 algorithm.
- Hash values are unique identifiers generated based on the contents of a file and can be used to verify data integrity.
- The `cmp` command compares files byte by byte and reports the differences found.
- Comparing hash values is an effective way to detect differences between files, even if their contents appear similar.
- Hash values provide a reliable method for validating data integrity and ensuring that files have not been tampered with.