

# CSC415-Device-Driver

## Build the kernel module and user application

You can build the kernel module by executing the following command in the Module directory.

```
make all
```

The user app can be built using the

```
make cryptographyTest
```

command in the Test directory.

You can clear all the build files using

```
make clean
```

command in both Module and Test directories.

## View the kernel logs

kernel logs of the Linux kernel can be seen by using

```
dmesg --following
```

command.

## Install the kernel module

You need to install the built kernel module into the kernel using

```
sudo insmod cryptography.ko
```

within the Module directory.

## Run the application

After installing the kernel module into the kernel, the user application can be run.

Use the sudo command to run the application, as we are going to access device files with the user application

```
sudo ./cryptographyTest
```

- First, you will be asked to provide a key. For that, you need to insert a single capital English letter.
- Then you will be asked to provide the mode. you need to insert the number of the option.
- Then insert the phrase you need to encrypt or decrypt.
- Finally, you will get the encrypted or decrypted phrase.

Note: as I'm using the symmetric key encryption technique, same key can be used for both encryption and decryption.

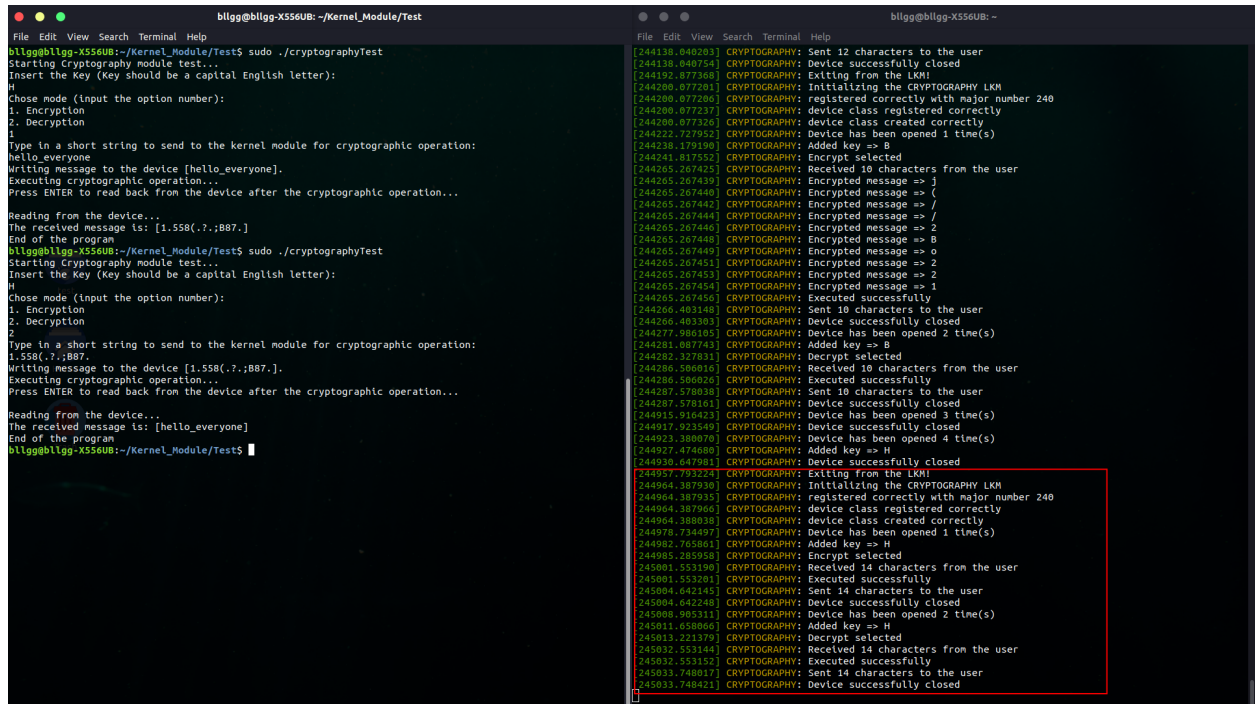
## Unload the kernel module

After running the user application, the kernel module should be unloaded.

```
sudo rmmod cryptography
```

can be used to unload the module.

# Screenshots



```
blgg@blgg-X550UB: ~/Kernel_Module/Test
File Edit View Search Terminal Help
blgg@blgg-X550UB:~/Kernel_Module/Test$ sudo ./cryptographyTest
Starting Cryptography module test...
Insert the Key (Key should be a capital English letter):
H
Chose mode (input the option number):
1. Encryption
2. Decryption
2
Type in a short string to send to the kernel module for cryptographic operation:
hello_everyone
Writing message to the device [hello_everyone].
Executing cryptographic operation...
Press ENTER to read back from the device after the cryptographic operation...

Reading from the device...
The received message is: [1.558(.?);B87.]
End of the program
blgg@blgg-X550UB:~/Kernel_Module/Test$ sudo ./cryptographyTest
Starting Cryptography module test...
Insert the Key (Key should be a capital English letter):
H
Chose mode (input the option number):
1. Encryption
2. Decryption
2
Type in a short string to send to the kernel module for cryptographic operation:
1.558(.?);B87.
Writing message to the device [1.558(.?);B87.].
Executing cryptographic operation...
Press ENTER to read back from the device after the cryptographic operation...

Reading from the device...
The received message is: [hello_everyone]
End of the program
blgg@blgg-X550UB:~/Kernel_Module/Test$
```

```
blgg@blgg-X550UB: ~
File Edit View Search Terminal Help
[244138.048083] CRYPTOGRAPHY: Sent 12 characters to the user
[244138.048754] CRYPTOGRAPHY: Device successfully closed
[244192.877368] CRYPTOGRAPHY: Exiting from the LKM!
[244280.077281] CRYPTOGRAPHY: Initializing the CRYPTOGRAPHY LKM
[244280.077284] CRYPTOGRAPHY: registered correctly with major number 240
[244280.077237] CRYPTOGRAPHY: device class registered correctly
[244280.077326] CRYPTOGRAPHY: device class created correctly
[244222.727952] CRYPTOGRAPHY: Device has been opened 1 time(s)
[244238.179198] CRYPTOGRAPHY: Added key => B
[244241.817552] CRYPTOGRAPHY: Encrypt selected
[244265.267425] CRYPTOGRAPHY: Received 10 characters from the user
[244265.267439] CRYPTOGRAPHY: Encrypted message => j
[244265.267446] CRYPTOGRAPHY: Encrypted message => (
[244265.267442] CRYPTOGRAPHY: Encrypted message => /
[244265.267444] CRYPTOGRAPHY: Encrypted message => /
[244265.267446] CRYPTOGRAPHY: Encrypted message => 2
[244265.267448] CRYPTOGRAPHY: Encrypted message => B
[244265.267449] CRYPTOGRAPHY: Encrypted message => 0
[244265.267451] CRYPTOGRAPHY: Encrypted message => 2
[244265.267453] CRYPTOGRAPHY: Encrypted message => 2
[244265.267454] CRYPTOGRAPHY: Encrypted message => 1
[244265.267456] CRYPTOGRAPHY: Executed successfully
[244266.483148] CRYPTOGRAPHY: Sent 10 characters to the user
[244266.483383] CRYPTOGRAPHY: Device successfully closed
[244277.986185] CRYPTOGRAPHY: Device has been opened 2 time(s)
[244281.087743] CRYPTOGRAPHY: Added key => B
[244282.327831] CRYPTOGRAPHY: Decrypt selected
[244286.586916] CRYPTOGRAPHY: Received 10 characters from the user
[244286.586926] CRYPTOGRAPHY: Executed successfully
[244287.578038] CRYPTOGRAPHY: Sent 10 characters to the user
[244287.578161] CRYPTOGRAPHY: Device successfully closed
[244915.916423] CRYPTOGRAPHY: Device has been opened 3 time(s)
[244917.923549] CRYPTOGRAPHY: Device successfully closed
[244923.388070] CRYPTOGRAPHY: Device has been opened 4 time(s)
[244927.474688] CRYPTOGRAPHY: Added key => H
[244930.647981] CRYPTOGRAPHY: Device successfully closed
[244957.793224] CRYPTOGRAPHY: Exiting from the LKM!
[244964.387930] CRYPTOGRAPHY: Initializing the CRYPTOGRAPHY LKM
[244964.387935] CRYPTOGRAPHY: registered correctly with major number 240
[244964.387966] CRYPTOGRAPHY: device class registered correctly
[244964.388038] CRYPTOGRAPHY: device class created correctly
[244978.734987] CRYPTOGRAPHY: Device has been opened 1 time(s)
[244982.765961] CRYPTOGRAPHY: Added key => H
[244985.285958] CRYPTOGRAPHY: Encrypt selected
[245081.553190] CRYPTOGRAPHY: Received 14 characters from the user
[245081.553261] CRYPTOGRAPHY: Executed successfully
[245084.642145] CRYPTOGRAPHY: Sent 14 characters to the user
[245084.642248] CRYPTOGRAPHY: Device successfully closed
[245088.985311] CRYPTOGRAPHY: Device has been opened 2 time(s)
[245011.458866] CRYPTOGRAPHY: Added key => H
[245013.221379] CRYPTOGRAPHY: Decrypt selected
[245032.553144] CRYPTOGRAPHY: Received 14 characters from the user
[245032.553152] CRYPTOGRAPHY: Executed successfully
[245033.748017] CRYPTOGRAPHY: Sent 14 characters to the user
[245033.748421] CRYPTOGRAPHY: Device successfully closed
```