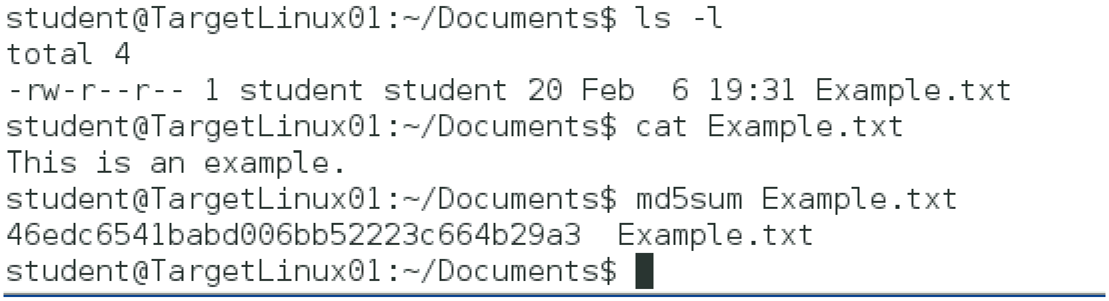
Eric Guzman 2/6/25
Hacking and Penetration Testing Professor Foti

Applying Encryption and Hashing Algorithms for Secure Communications

SECTION 1: Hands-On Demonstration

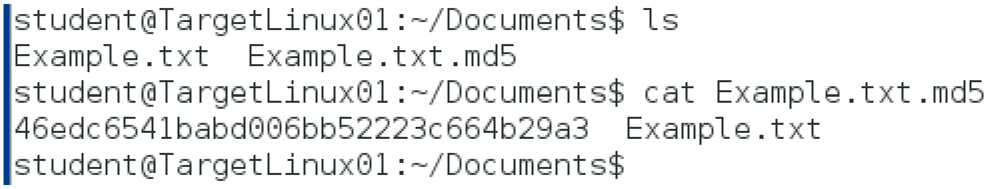
Part 2: Create an MD5sum and a SHA1sum Hash String

7. Make a screen capture showing the MD5sum hash string and paste it into your Lab Report file.



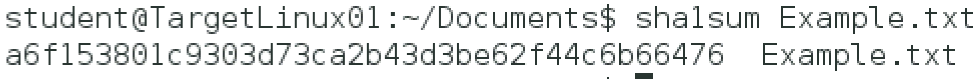
```
student@TargetLinux01:~/Documents$ ls -l
total 4
-rw-r--r-- 1 student student 20 Feb  6 19:31 Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example.
student@TargetLinux01:~/Documents$ md5sum Example.txt
46edc6541babd006bb52223c664b29a3  Example.txt
student@TargetLinux01:~/Documents$
```

11. Make a screen capture showing the contents of Example.txt.md5 file and paste it into your Lab Report file.



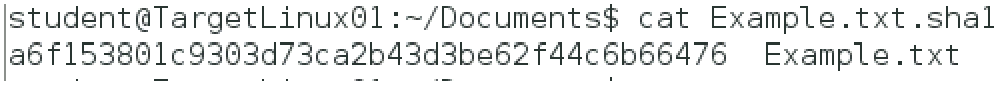
```
student@TargetLinux01:~/Documents$ ls
Example.txt  Example.txt.md5
student@TargetLinux01:~/Documents$ cat Example.txt.md5
46edc6541babd006bb52223c664b29a3  Example.txt
student@TargetLinux01:~/Documents$
```

15. Make a screen capture showing the SHA1sum hash string and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ sha1sum Example.txt
a6f153801c9303d73ca2b43d3be62f44c6b66476  Example.txt
```

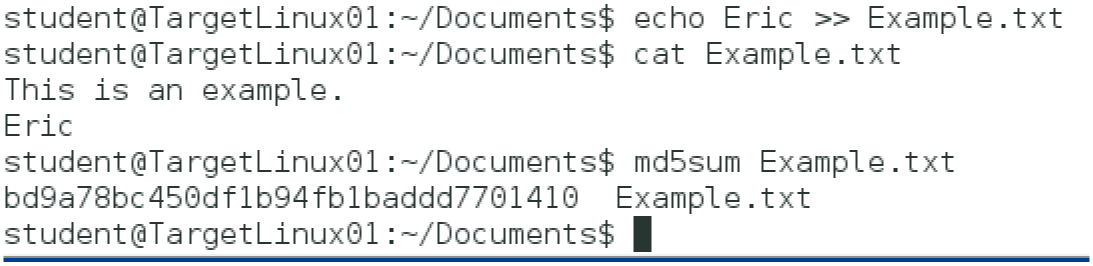
19. Make a screen capture showing the contents of Example.txt.sha1 file and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ cat Example.txt.sha1
a6f153801c9303d73ca2b43d3be62f44c6b66476  Example.txt
```

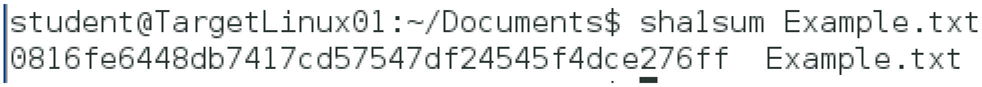
Part 3: Modify a File and Verify Hash Values

4. Make a screen capture showing the new MD5sum hash string and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ echo Eric >> Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example.
Eric
student@TargetLinux01:~/Documents$ md5sum Example.txt
bd9a78bc450df1b94fb1baddd7701410  Example.txt
student@TargetLinux01:~/Documents$ 
```

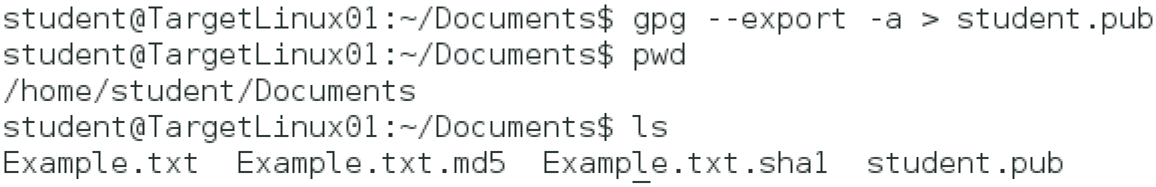
6. Make a screen capture showing the new SHA1 hash string and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ sha1sum Example.txt
0816fe6448db7417cd57547df24545f4dce276ff  Example.txt
```

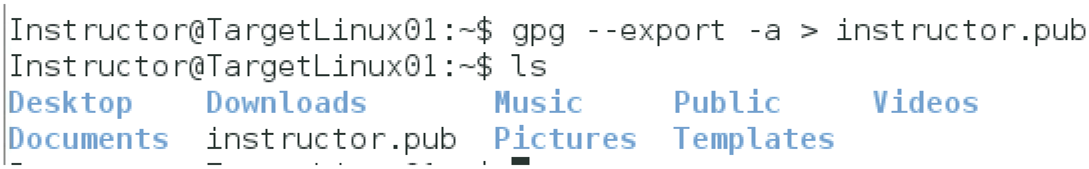
Part 4: Generate GnuPG Keys

13. Make a screen capture showing the contents of the /home/student/Documents folder and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ gpg --export -a > student.pub
student@TargetLinux01:~/Documents$ pwd
/home/student/Documents
student@TargetLinux01:~/Documents$ ls
Example.txt  Example.txt.md5  Example.txt.sha1  student.pub
```

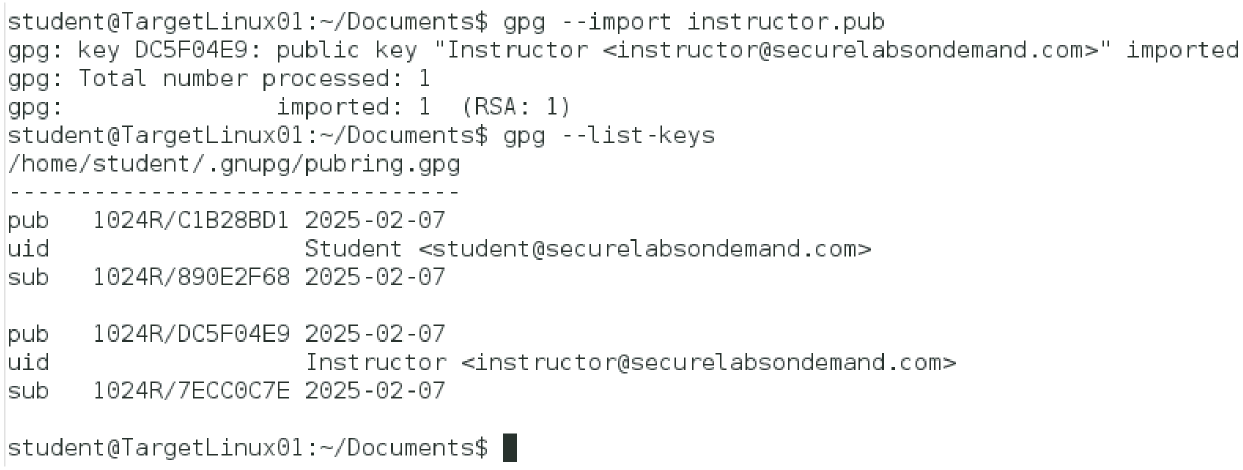
21. Make a screen capture showing the contents of the /home/instructor folder and paste it into your Lab Report file.



```
Instructor@TargetLinux01:~$ gpg --export -a > instructor.pub
Instructor@TargetLinux01:~$ ls
Desktop     Downloads       Music     Public     Videos
Documents   instructor.pub  Pictures  Templates
```

Part 5: Share a GnuPG Key

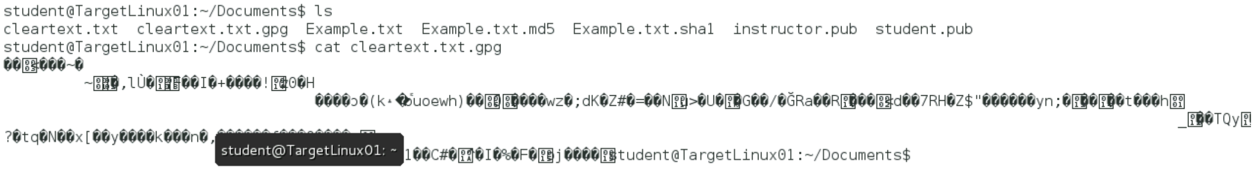
6. Make a screen capture showing the student’s public key ring and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ gpg --import instructor.pub
gpg: key DC5F04E9: public key "Instructor <instructor@securelabsondemand.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
student@TargetLinux01:~/Documents$ gpg --list-keys
/home/student/.gnupg/pubring.gpg
--------------------------------
pub   1024R/C1B28BD1 2025-02-07
uid                  Student <student@securelabsondemand.com>
sub   1024R/890E2F68 2025-02-07

pub   1024R/DC5F04E9 2025-02-07
uid                  Instructor <instructor@securelabsondemand.com>
sub   1024R/7ECC0C7E 2025-02-07

student@TargetLinux01:~/Documents$ 
```

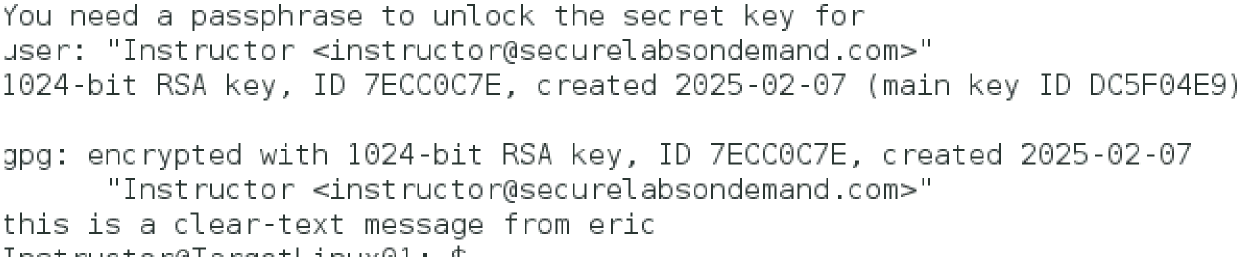
Part 6: Encrypt and Decrypt a ClearText Message

8. Make a screen capture showing the contents of the encrypted file and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ ls
cleartext.txt  cleartext.txt.gpg  Example.txt  Example.txt.md5  Example.txt.sha1  instructor.pub  student.pub
student@TargetLinux01:~/Documents$ cat cleartext.txt.gpg
��▯���~�
        ~▯▯�,lÙ�▯▯��I�+����!▯0�H
                                                   ����ɔ�(k·�uoewh)��▯▯����wz�;dK�Z#�=��N▯>�U�▯G��/�ĞRa��R▯���▯d��7RH�Z$"������yn;�▯��▯�t���h▯
                                                                                                                                  _▯��TQy▯
?�tq�N��x[��y����k���n�,
1��C#�▯�I�%�F�▯j����▯tudent@TargetLinux01:~/Documents$
```

19. Make a screen capture showing the contents of the decrypted cleartext.txt.gpg file and paste it into your Lab Report file.

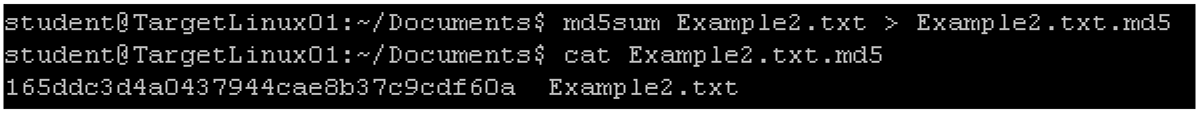


```
You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 7ECC0C7E, created 2025-02-07 (main key ID DC5F04E9)

gpg: encrypted with 1024-bit RSA key, ID 7ECC0C7E, created 2025-02-07
      "Instructor <instructor@securelabsondemand.com>"
this is a clear-text message from eric
Instructor@TargetLinux01:~$
```

SECTION 2: Applied Learning

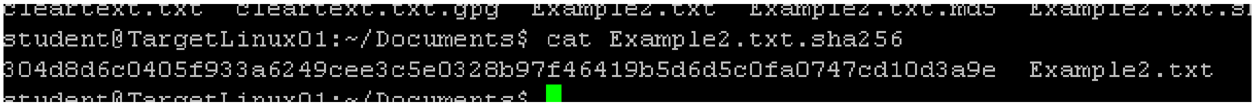
Part 2: Create an MD5sum and a SHA256sum Hash String

6. Make a screen capture showing the contents of Example2.txt.md5 file and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ md5sum Example2.txt > Example2.txt.md5
student@TargetLinux01:~/Documents$ cat Example2.txt.md5
165ddc3d4a0437944cae8b37c9cdf60a  Example2.txt
```

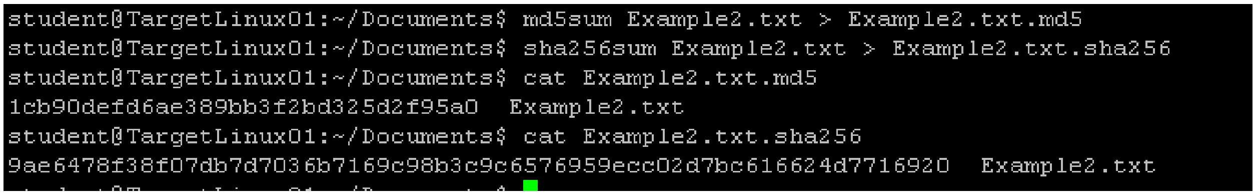
13. Make a screen capture showing the contents of Example2.txt.sha256 file and paste it into your Lab Report file.



```
cleartext.txt  cleartext.txt.gpg  Example2.txt  Example2.txt.md5  Example2.txt.s
student@TargetLinux01:~/Documents$ cat Example2.txt.sha256
304d8d6c0405f933a6249cee3c5e0328b97f46419b5d6d5c0fa0747cd10d3a9e  Example2.txt
student@TargetLinux01:~/Documents$
```

Part 3: Modify a File and Verify Hash Values

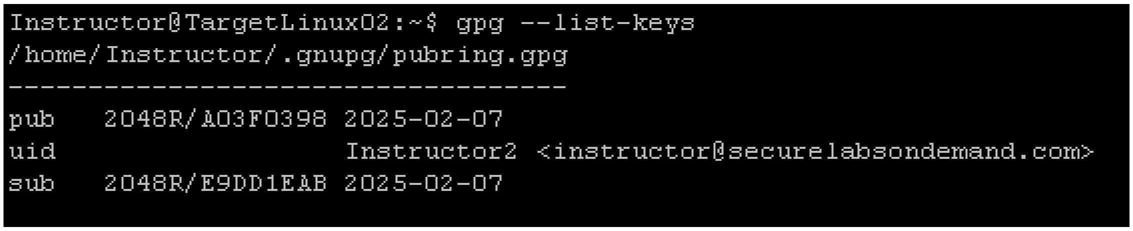
7. Make a screen capture showing the modified MD5sum and SHA-256 hash strings and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ md5sum Example2.txt > Example2.txt.md5
student@TargetLinux01:~/Documents$ sha256sum Example2.txt > Example2.txt.sha256
student@TargetLinux01:~/Documents$ cat Example2.txt.md5
1cb90defd6ae389bb3f2bd325d2f95a0  Example2.txt
student@TargetLinux01:~/Documents$ cat Example2.txt.sha256
9ae6478f38f07db7d7036b7169c98b3c9c6576959ecc02d7bc616624d7716920  Example2.txt
student@TargetLinux01:~/Documents$
```

Part 4: Generate GnuPG Keys

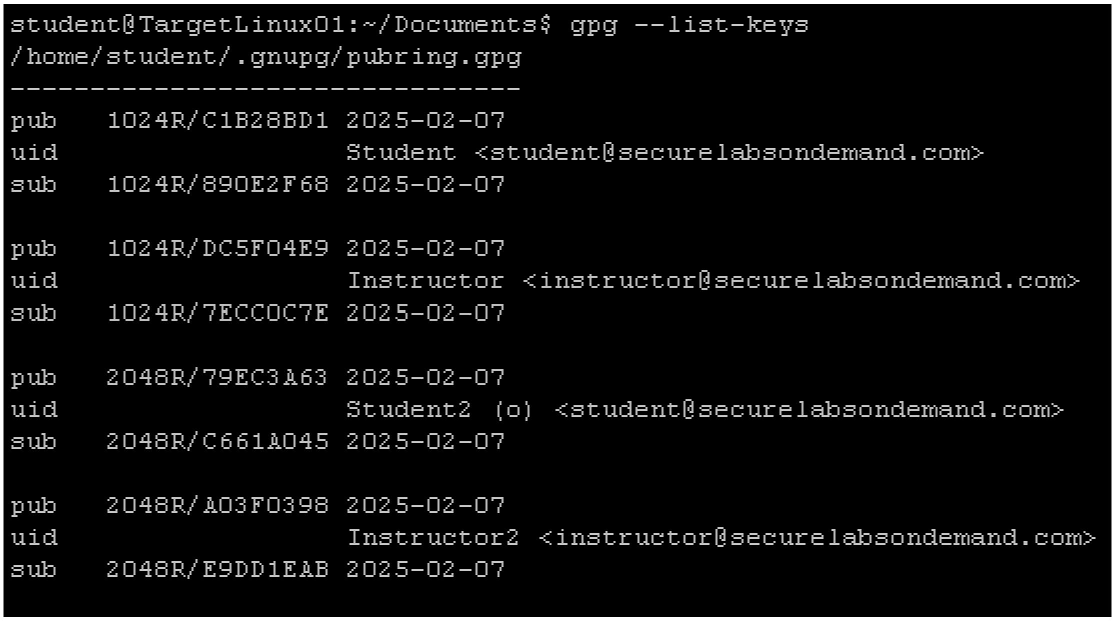
17. Make a screen capture showing the instructor’s public key ring and paste it into your Lab Report file.



```
Instructor@TargetLinux02:~$ gpg --list-keys
/home/Instructor/.gnupg/pubring.gpg
-----------------------------------
pub   2048R/A03F0398 2025-02-07
uid                  Instructor2 <instructor@securelabsondemand.com>
sub   2048R/E9DD1EAB 2025-02-07
```

Part 5: Share a GnuPG Key

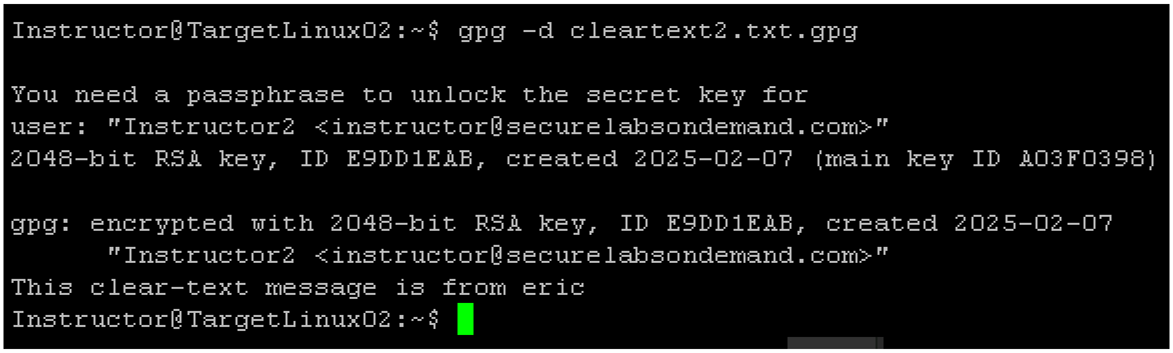
12. Make a screen capture showing the student’s public key ring and paste it into your Lab Report file.



```
student@TargetLinux01:~/Documents$ gpg --list-keys
/home/student/.gnupg/pubring.gpg
--------------------------------
pub   1024R/C1B28BD1 2025-02-07
uid                  Student <student@securelabsondemand.com>
sub   1024R/890E2F68 2025-02-07

pub   1024R/DC5F04E9 2025-02-07
uid                  Instructor <instructor@securelabsondemand.com>
sub   1024R/7ECC0C7E 2025-02-07

pub   2048R/79EC3A63 2025-02-07
uid                  Student2 (o) <student@securelabsondemand.com>
sub   2048R/C661A045 2025-02-07

pub   2048R/A03F0398 2025-02-07
uid                  Instructor2 <instructor@securelabsondemand.com>
sub   2048R/E9DD1EAB 2025-02-07
```

Part 6: Encrypt and Decrypt a ClearText Message

19. Make a screen capture showing the contents of the decrypted cleartext2.txt.gpg file and paste it into your Lab Report file.



```
Instructor@TargetLinux02:~$ gpg -d cleartext2.txt.gpg

You need a passphrase to unlock the secret key for
user: "Instructor2 <instructor@securelabsondemand.com>"
2048-bit RSA key, ID E9DD1EAB, created 2025-02-07 (main key ID A03F0398)

gpg: encrypted with 2048-bit RSA key, ID E9DD1EAB, created 2025-02-07
      "Instructor2 <instructor@securelabsondemand.com>"
This clear-text message is from eric
Instructor@TargetLinux02:~$ 
```

SECTION 3: Lab Challenge and Analysis

Part 1: Analysis and Discussion

- Describe the differences between RSA and ECDSA encryption algorithms, and name a well-known product that uses each type of encryption. Cite your references.

RSA and ECDSA are both pretty popular cryptographic algorithms, but differ in how they are implemented and how efficient they are. RSA focuses on how difficult it is to factor large prime numbers where ECDSA is based around the ECDLP which allows it to have a similar security level while having much smaller key sizes. RSA takes longer to generate its keys and signing but is faster for verification. ECDSA is the opposite, having quicker key generation and signing but slower verification. RSA can be found in SSL, TLS certificates and OpenVPN, where ECDSA is found in the blockchain and SSH authentication.

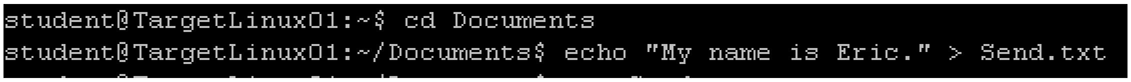
References:

1. NIST. (2020). FIPS 186-4: Digital Signature Standard (DSS). National Institute of Standards and Technology
2. Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley.

Part 2: Tools and Commands

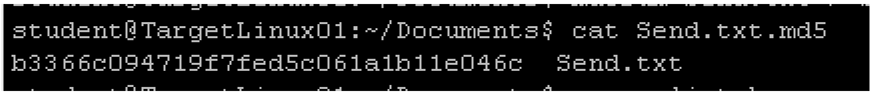
On your own, create a new text file called Send.txt, apply an MD5 hash to the file, and then use GnuPG to encrypt and decrypt it. In your Challenge Questions file, include screen captures to document the steps you followed.

a. Create a new file called Send.txt and add the following text: My name is yourname, replacing yourname with your own name. (Hint: Refer to Part 1 of this lab).



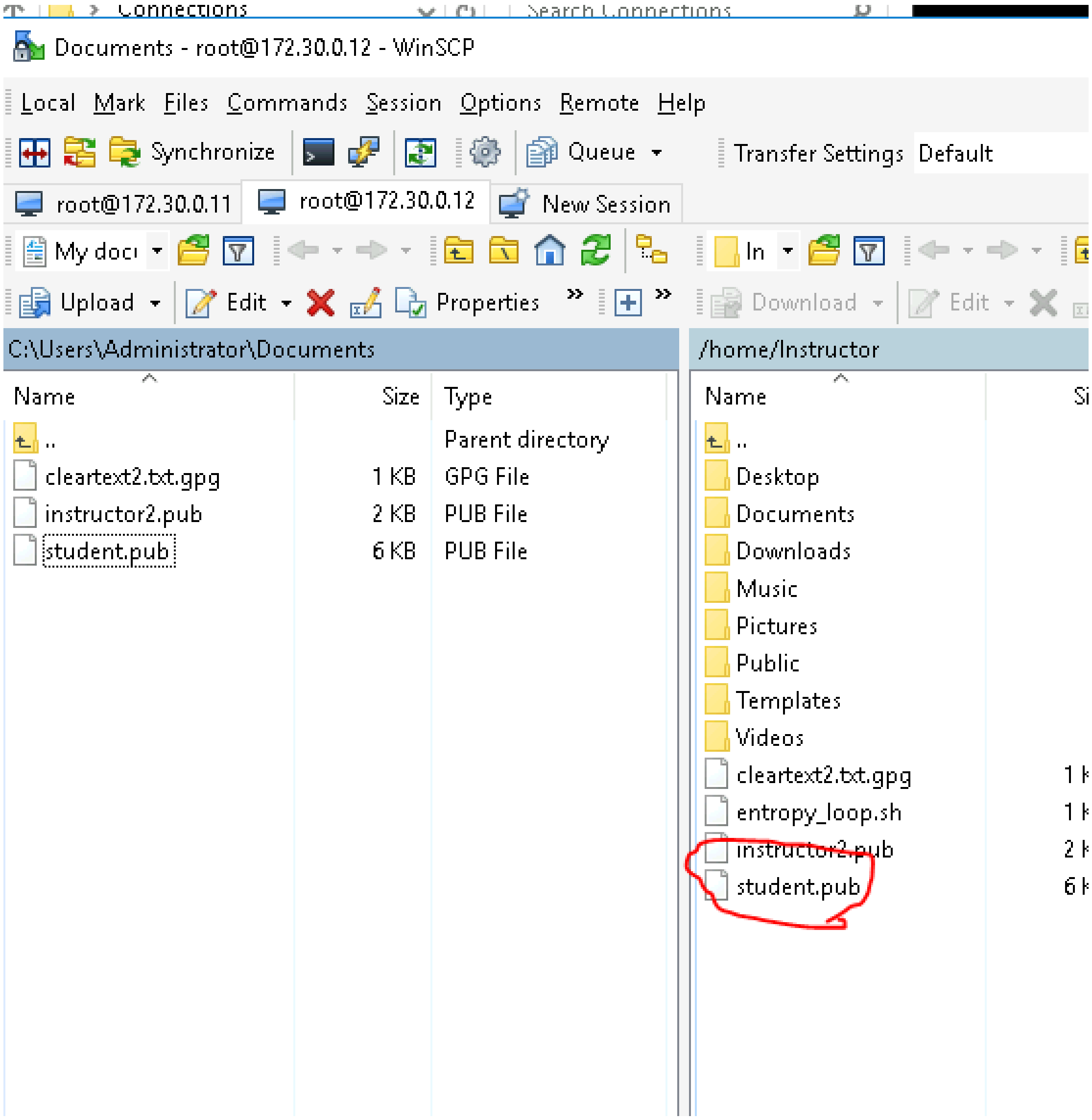
```
student@TargetLinux01:~$ cd Documents
student@TargetLinux01:~/Documents$ echo "My name is Eric." > Send.txt
```

b. Add an MD5sum hash to the file and record the string in your Challenge Questions file (Hint: Refer to Part 2 of this lab).



```
student@TargetLinux01:~/Documents$ cat Send.txt.md5
b3366c094719f7fed5c061a1b11e046c  Send.txt
```

c. Share the student's public key with the instructor account, so the instructor can send encrypted messages to the student account (Hint: Refer to Part 5 of this lab).



Documents - root@172.30.0.12 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
root@172.30.0.11
root@172.30.0.12
New Session
My doc
In
Upload
Edit
Properties
Download
Edit

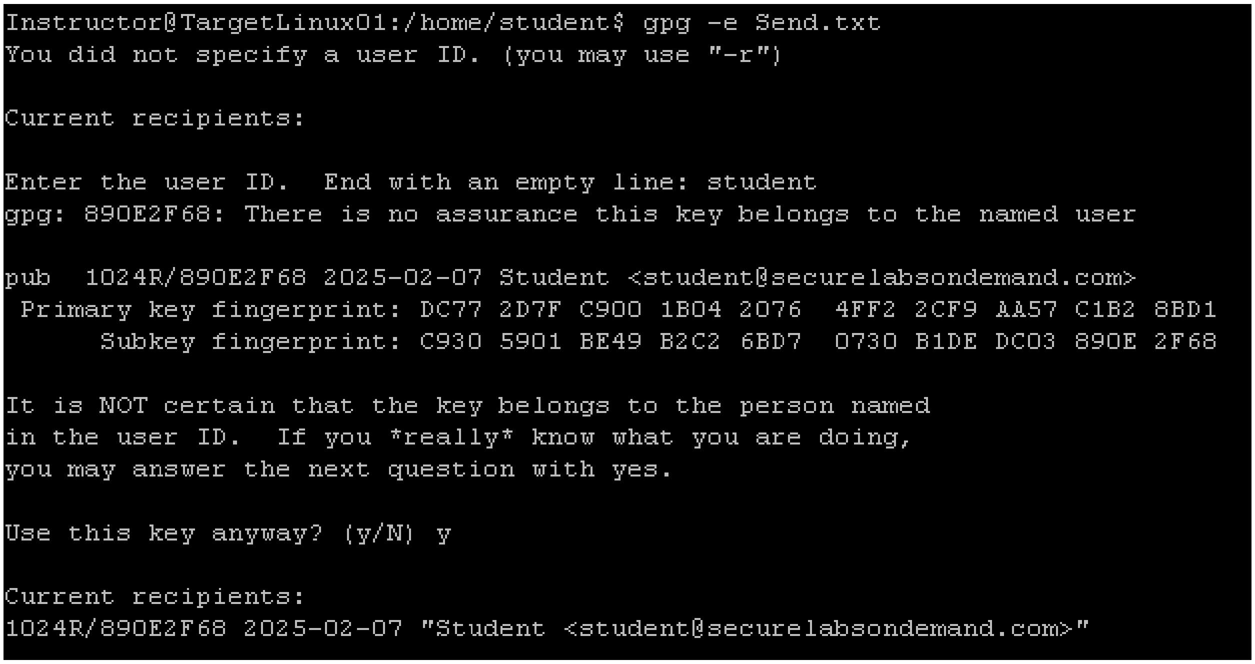
C:\Users\Administrator\Documents

Name	Size	Type
..		Parent directory
cleartext2.txt.gpg	1 KB	GPG File
instructor2.pub	2 KB	PUB File
student.pub	6 KB	PUB File

/home/Instructor

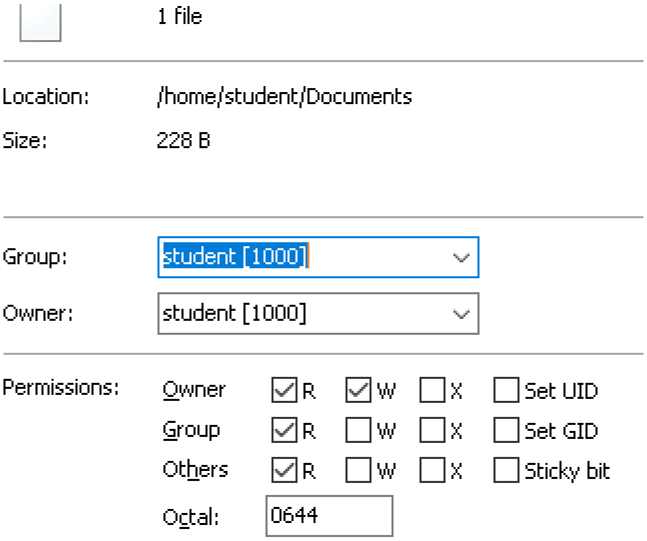
Name	Si
..	
Desktop	
Documents	
Downloads	
Music	
Pictures	
Public	
Templates	
Videos	
cleartext2.txt.gpg	1 k
entropy_loop.sh	1 k
instructor2.pub	2 k
student.pub	6 k

d. Using the instructor account, encrypt Send.txt using GnuPG (Hint: Refer to Part 6 of this lab).



```
Instructor@TargetLinux01:/home/student$ gpg -e Send.txt
You did not specify a user ID. (you may use "-r")

Current recipients:

Enter the user ID.  End with an empty line: student
gpg: 890E2F68: There is no assurance this key belongs to the named user

pub  1024R/890E2F68 2025-02-07 Student <student@securelabsondemand.com>
 Primary key fingerprint: DC77 2D7F C900 1B04 2076  4FF2 2CF9 AA57 C1B2 8BD1
      Subkey fingerprint: C930 5901 BE49 B2C2 6BD7  0730 B1DE DC03 890E 2F68

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y

Current recipients:
1024R/890E2F68 2025-02-07 "Student <student@securelabsondemand.com>"
```

e. Copy the file to the student's home folder, change the permissions on the file, and then decrypt Send.txt using the student account.



1 file

Location: /home/student/Documents

Size: 228 B

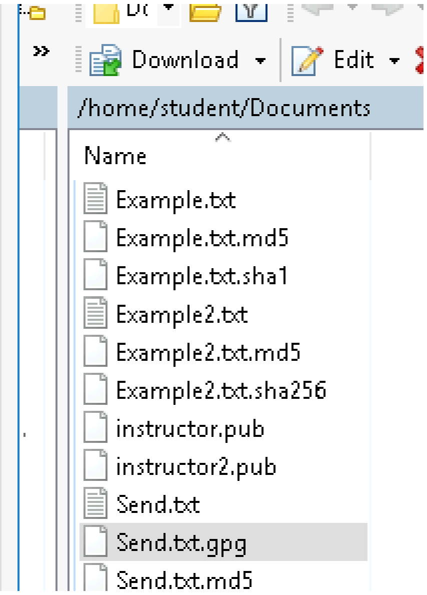
Group: student [1000]

Owner: student [1000]

Permissions:

Owner	☑ R	☑ W	□ X	□ Set UID
Group	☑ R	□ W	□ X	□ Set GID
Others	☑ R	□ W	□ X	□ Sticky bit

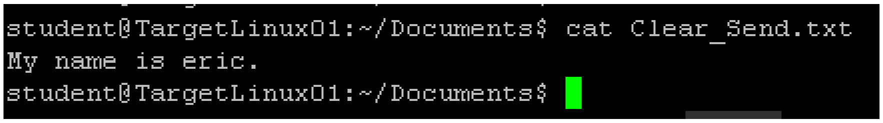
Octal: 0644



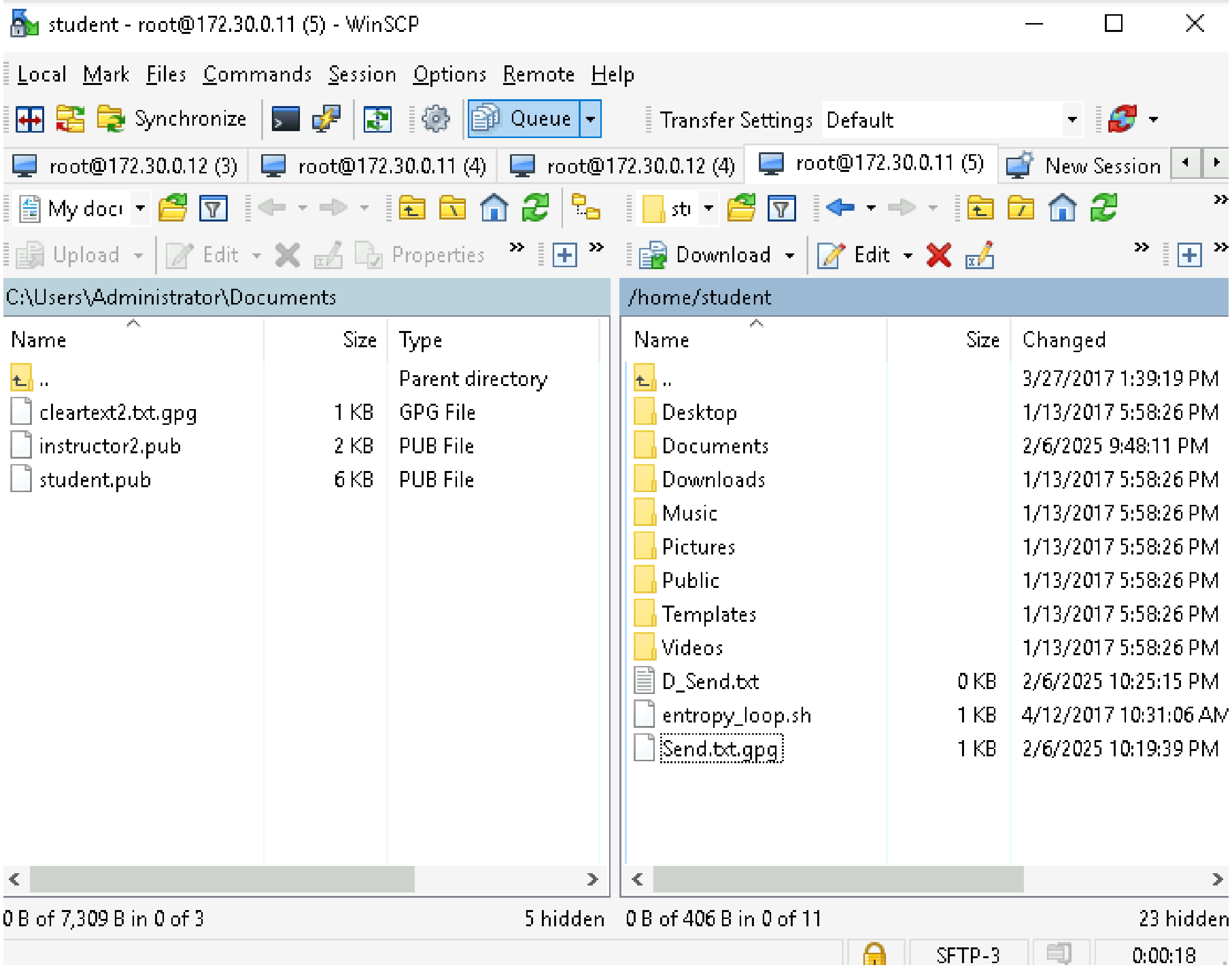
/home/student/Documents

Name

Example.txt
Example.txt.md5
Example.txt.sha1
Example2.txt
Example2.txt.md5
Example2.txt.sha256
instructor.pub
instructor2.pub
Send.txt
Send.txt.gpg
Send.txt.md5



```
student@TargetLinux01:~/Documents$ cat Clear_Send.txt
My name is eric.
student@TargetLinux01:~/Documents$
```



C:\Users\Administrator\Documents

Name	Size	Type
..		Parent directory
cleartext2.txt.gpg	1 KB	GPG File
instructor2.pub	2 KB	PUB File
student.pub	6 KB	PUB File

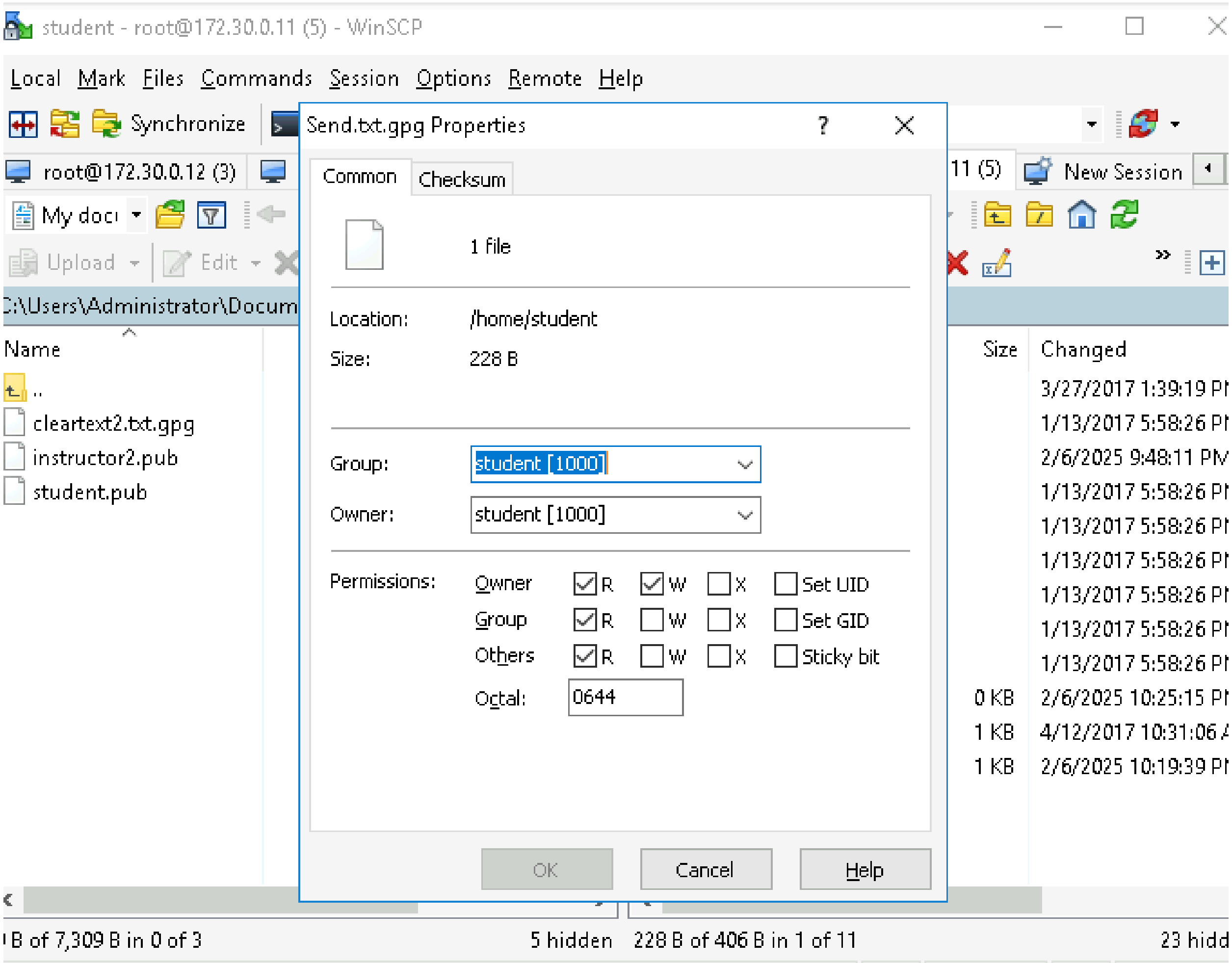
/home/student

Name	Size	Changed
..		3/27/2017 1:39:19 PM
Desktop		1/13/2017 5:58:26 PM
Documents		2/6/2025 9:48:11 PM
Downloads		1/13/2017 5:58:26 PM
Music		1/13/2017 5:58:26 PM
Pictures		1/13/2017 5:58:26 PM
Public		1/13/2017 5:58:26 PM
Templates		1/13/2017 5:58:26 PM
Videos		1/13/2017 5:58:26 PM
D_Send.txt	0 KB	2/6/2025 10:25:15 PM
entropy_loop.sh	1 KB	4/12/2017 10:31:06 AM
Send.txt.gpg	1 KB	2/6/2025 10:19:39 PM

0 B of 7,309 B in 0 of 3 5 hidden

0 B of 406 B in 0 of 11 23 hidden

SFTP-3 0:00:18



student - root@172.30.0.11 (5) - WinSCP

Local Mark Files Commands Session Options Remote Help

Send.txt.gpg Properties

Common | Checksum

1 file

Location: /home/student

Size: 228 B

Group: student [1000]

Owner: student [1000]

Permissions:

Owner	☑ R	☑ W	□ X	□ Set UID
Group	☑ R	□ W	□ X	□ Set GID
Others	☑ R	□ W	□ X	□ Sticky bit

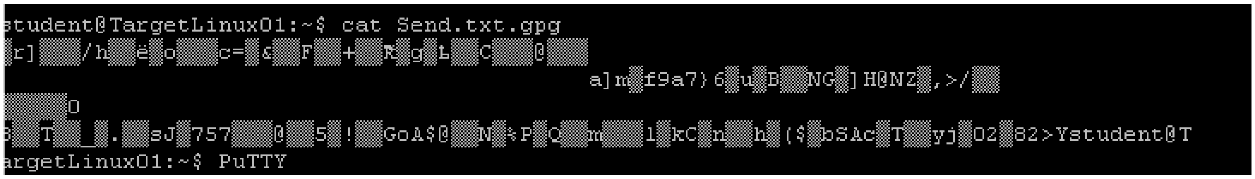
Octal: 0644

OK Cancel Help

Size	Changed
	3/27/2017 1:39:19 PI
	1/13/2017 5:58:26 PI
	2/6/2025 9:48:11 PM
	1/13/2017 5:58:26 PI
	1/13/2017 5:58:26 PI
	1/13/2017 5:58:26 PI
	1/13/2017 5:58:26 PI
	1/13/2017 5:58:26 PI
	1/13/2017 5:58:26 PI
0 KB	2/6/2025 10:25:15 PI
1 KB	4/12/2017 10:31:06 A
1 KB	2/6/2025 10:19:39 PI

B of 7,309 B in 0 of 3 5 hidden 228 B of 406 B in 1 of 11 23 hidd

f. In your Challenge Questions file, include screen captures of the commands used to encrypt and decrypt the Send.txt file.

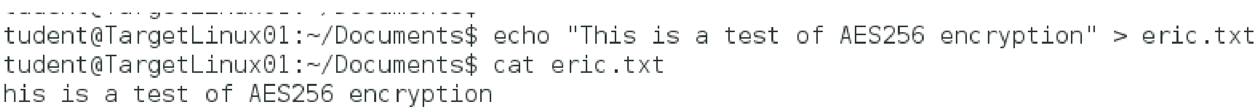


```
student@TargetLinux01:~$ cat Send.txt.gpg
r]▒▒▒/h▒▒e▒o▒▒▒c=▒&▒F▒▒+▒▒R▒g▒L▒▒C▒▒▒@▒▒▒
                                            a]m▒f9a7)6▒u▒B▒▒NG▒]H@NZ▒,>/▒▒
▒▒▒▒▒0
3▒▒T▒▒_▒.▒▒sJ▒757▒▒▒@▒▒5▒!▒▒GoA$@▒▒N▒%P▒Q▒▒m▒▒▒1▒kC▒n▒▒h▒($▒bSAc▒T▒▒yj▒O2▒82>Ystudent@T
argetLinux01:~$ PuTTY
```

Part 3: Challenge Exercise

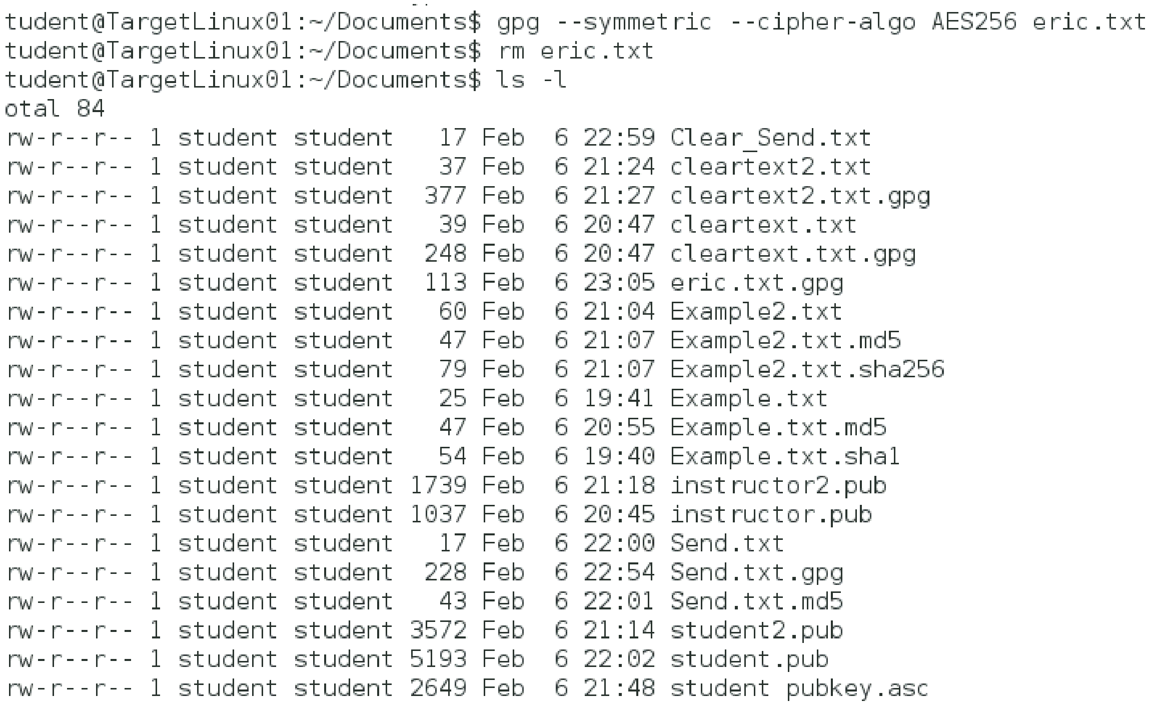
Not all encryption methods require an encryption key pair. AES256 requires only a shared passphrase and the correct syntax. Use the Internet to research the correct syntax for using this encryption method with GPG and complete the following tasks to encrypt and decrypt a new file. Make screen captures to record your progress.

a. On TargetLinux01, log in as the student and create a new file called yourname.txt, replacing yourname with your own name, and add the following text: This is a test of AES256 encryption.



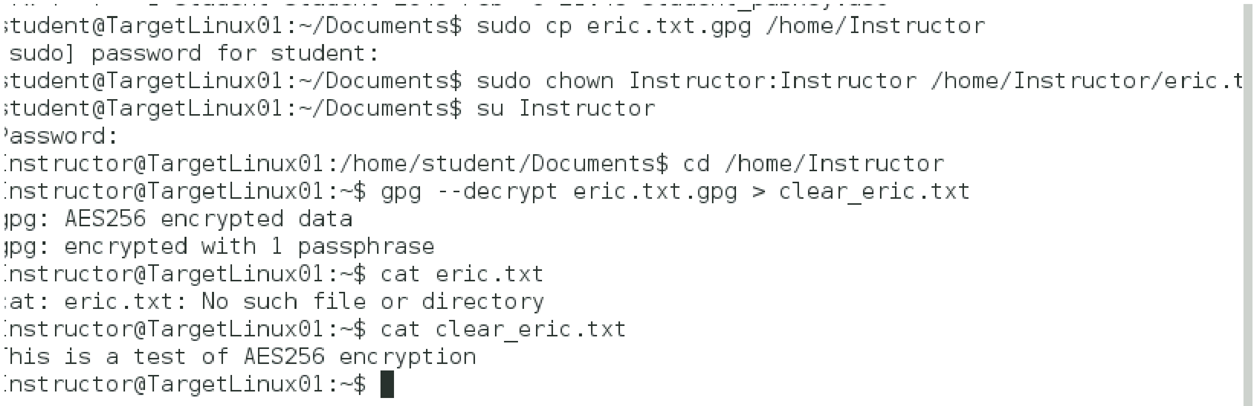
```
tudent@TargetLinux01:~/Documents$ echo "This is a test of AES256 encryption" > eric.txt
tudent@TargetLinux01:~/Documents$ cat eric.txt
his is a test of AES256 encryption
```

b. Use AES256 encryption with GPG to encrypt the file and remove the original file from the directory, leaving only the encrypted version of the file.



```
tudent@TargetLinux01:~/Documents$ gpg --symmetric --cipher-algo AES256 eric.txt
tudent@TargetLinux01:~/Documents$ rm eric.txt
tudent@TargetLinux01:~/Documents$ ls -l
otal 84
rw-r--r-- 1 student student   17 Feb  6 22:59 Clear_Send.txt
rw-r--r-- 1 student student   37 Feb  6 21:24 cleartext2.txt
rw-r--r-- 1 student student  377 Feb  6 21:27 cleartext2.txt.gpg
rw-r--r-- 1 student student   39 Feb  6 20:47 cleartext.txt
rw-r--r-- 1 student student  248 Feb  6 20:47 cleartext.txt.gpg
rw-r--r-- 1 student student  113 Feb  6 23:05 eric.txt.gpg
rw-r--r-- 1 student student   60 Feb  6 21:04 Example2.txt
rw-r--r-- 1 student student   47 Feb  6 21:07 Example2.txt.md5
rw-r--r-- 1 student student   79 Feb  6 21:07 Example2.txt.sha256
rw-r--r-- 1 student student   25 Feb  6 19:41 Example.txt
rw-r--r-- 1 student student   47 Feb  6 20:55 Example.txt.md5
rw-r--r-- 1 student student   54 Feb  6 19:40 Example.txt.sha1
rw-r--r-- 1 student student 1739 Feb  6 21:18 instructor2.pub
rw-r--r-- 1 student student 1037 Feb  6 20:45 instructor.pub
rw-r--r-- 1 student student   17 Feb  6 22:00 Send.txt
rw-r--r-- 1 student student  228 Feb  6 22:54 Send.txt.gpg
rw-r--r-- 1 student student   43 Feb  6 22:01 Send.txt.md5
rw-r--r-- 1 student student 3572 Feb  6 21:14 student2.pub
rw-r--r-- 1 student student 5193 Feb  6 22:02 student.pub
rw-r--r-- 1 student student 2649 Feb  6 21:48 student pubkey.asc
```

c. Copy the file to the Instructor account, change the permissions on the file, and then decrypt the file using the encryption passphrase.



```
student@TargetLinux01:~/Documents$ sudo cp eric.txt.gpg /home/Instructor
 sudo] password for student:
student@TargetLinux01:~/Documents$ sudo chown Instructor:Instructor /home/Instructor/eric.t
student@TargetLinux01:~/Documents$ su Instructor
Password:
Instructor@TargetLinux01:/home/student/Documents$ cd /home/Instructor
Instructor@TargetLinux01:~$ gpg --decrypt eric.txt.gpg > clear_eric.txt
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
Instructor@TargetLinux01:~$ cat eric.txt
cat: eric.txt: No such file or directory
Instructor@TargetLinux01:~$ cat clear_eric.txt
This is a test of AES256 encryption
Instructor@TargetLinux01:~$ █
```