LAB7: Applying Encryption and Hashing Algorithms for Secure Communications

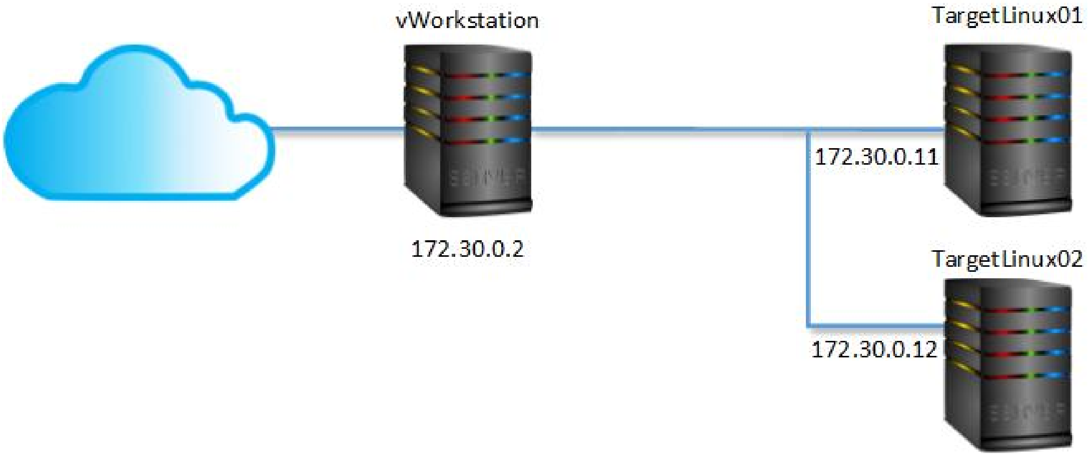
Section 3: Lab Challenge and Analysis

Table of Contents

Topology

This lab contains the following virtual devices. Please refer to the network topology diagram below.

- vWorkstation (Windows Server 2016)
- TargetLinux01 (Debian Linux)
- TargetLinux02 (Debian Linux)



Tools and Software

The following software and/or utilities are required to complete this lab. Students are encouraged to explore the Internet to learn more about the products and tools used in this lab.

- GNU Privacy Guard (GnuPG or GPG)
- KeyTransfer
- WinSCP
- vi Editor

Part 1: Analysis and Discussion

Two popular asymmetric encryption algorithms with different properties and applications are RSA (Rivest-Shamir-Adleman) and ECDSA (Elliptic Curve Digital Signature Algorithm).

RSA Encryption:

1. Key Length: For the same security levels, RSA typically employs longer key lengths than ECDSA. Common key lengths are 2048 bits or 4096 bits.
2. Computational Complexity: RSA encryption and decryption operations can be computationally intensive, especially with longer key lengths.
3. Security: The difficulty of factoring enormous integers forms the basis of RSA's security. However, because of advancements in computing power and factoring algorithms, longer key lengths are required for good security.
4. Key Generation: Finding large prime numbers during key generation for RSA can take some time.
5. Usage: RSA is commonly used for tasks like secure email (PGP), SSL/TLS encryption for web communication, and digital signatures.

A well-known product that uses RSA encryption is OpenSSL, a widely used open-source toolkit for implementing the SSL/TLS protocol that includes RSA for encryption and digital signatures.

ECDSA Encryption:

1. Key Length: ECDSA provides equivalent security to RSA with much shorter key lengths. A typical ECDSA key length might be 256 bits.
2. Computational Complexity: ECDSA operations are faster and require fewer computational resources compared to RSA, making it suitable for resource-constrained environments.
3. Security: ECDSA's security is based on the elliptic curve discrete logarithm problem, which is more resistant to quantum attacks than RSA's factorization problem.
4. Key Generation: Key generation for ECDSA is faster than RSA since it involves operations on elliptic curves.
5. Usage: ECDSA is commonly used in modern cryptographic protocols and systems where efficiency and security are critical, such as in blockchain technologies (e.g., Bitcoin) and secure messaging apps.

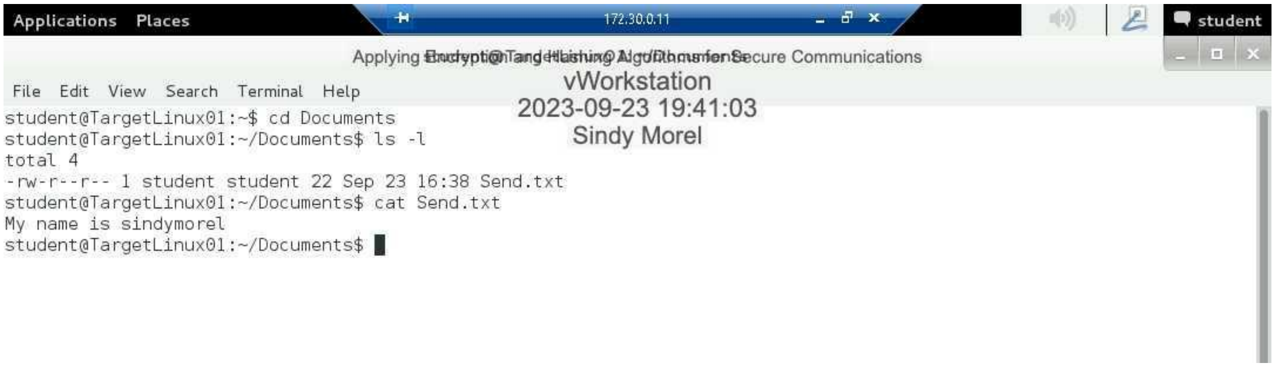
Popular item employing ECDSA encryption: Since it offers quick and safe digital signatures, ECDSA is used for transaction signing by Bitcoin and many other cryptocurrencies. The decision between RSA and ECDSA frequently depends on the needs of a system, including the required level of security, the available computational resources, and the use case.

References:

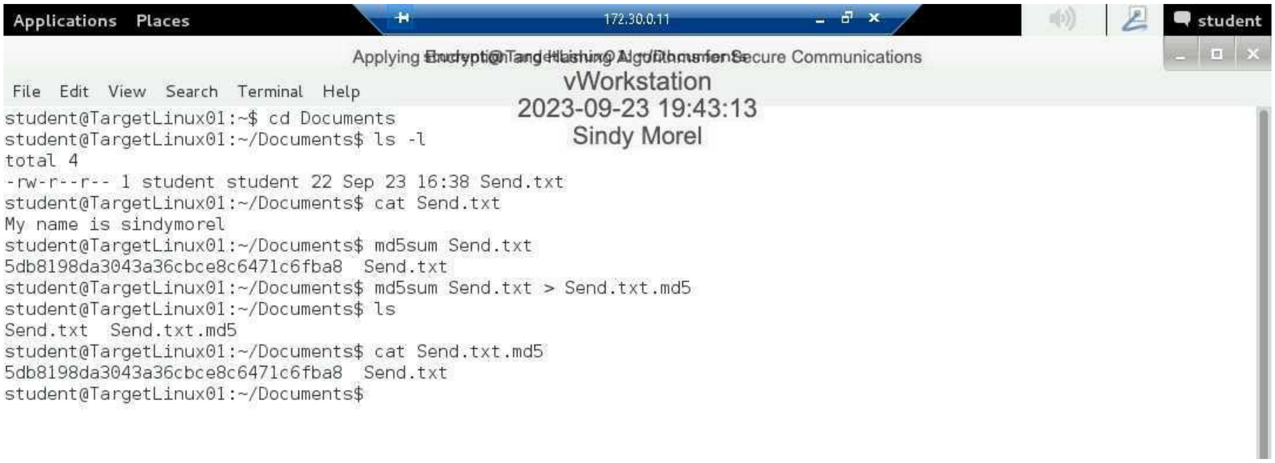
Stinson, D. R. (2006). Cryptography: Theory and Practice (3rd ed.). CRC Press.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.

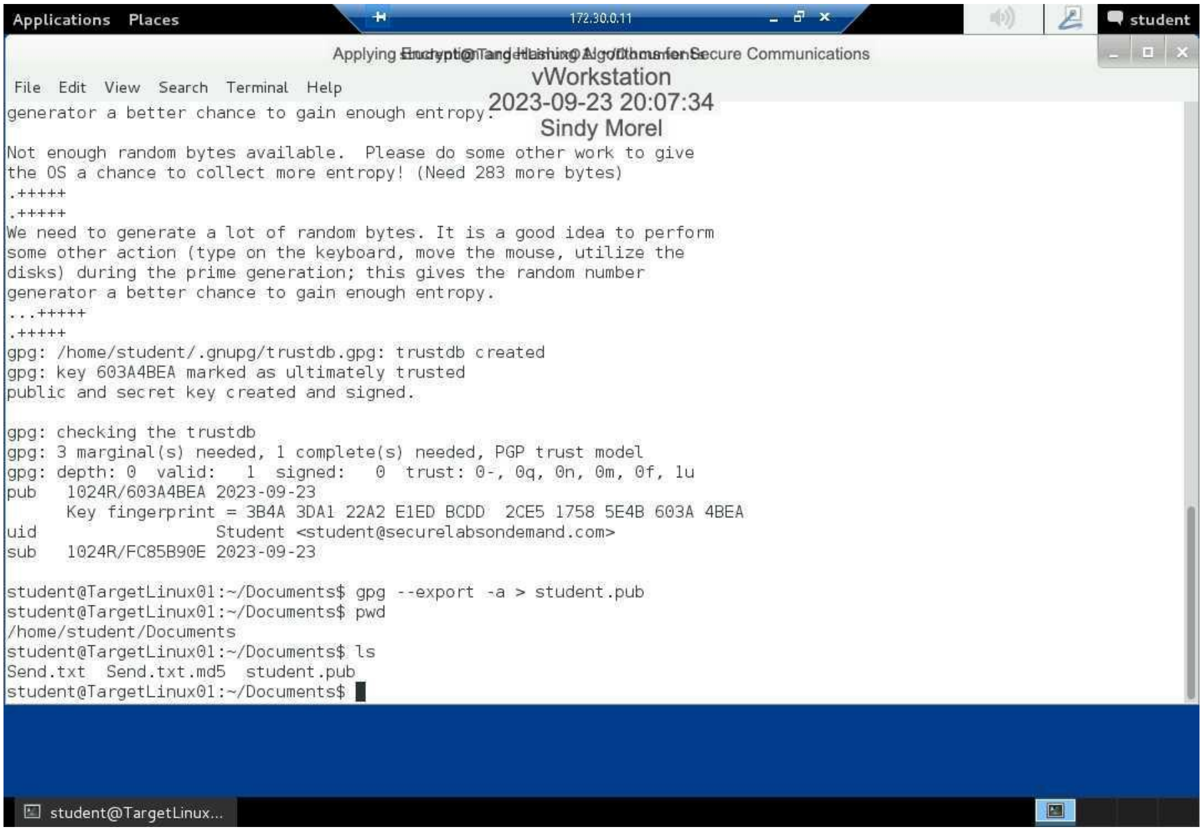
Part 2: Tools and Commands



```
student@TargetLinux01:~$ cd Documents
student@TargetLinux01:~/Documents$ ls -l
total 4
-rw-r--r-- 1 student student 22 Sep 23 16:38 Send.txt
student@TargetLinux01:~/Documents$ cat Send.txt
My name is sindymorel
student@TargetLinux01:~/Documents$ 
```

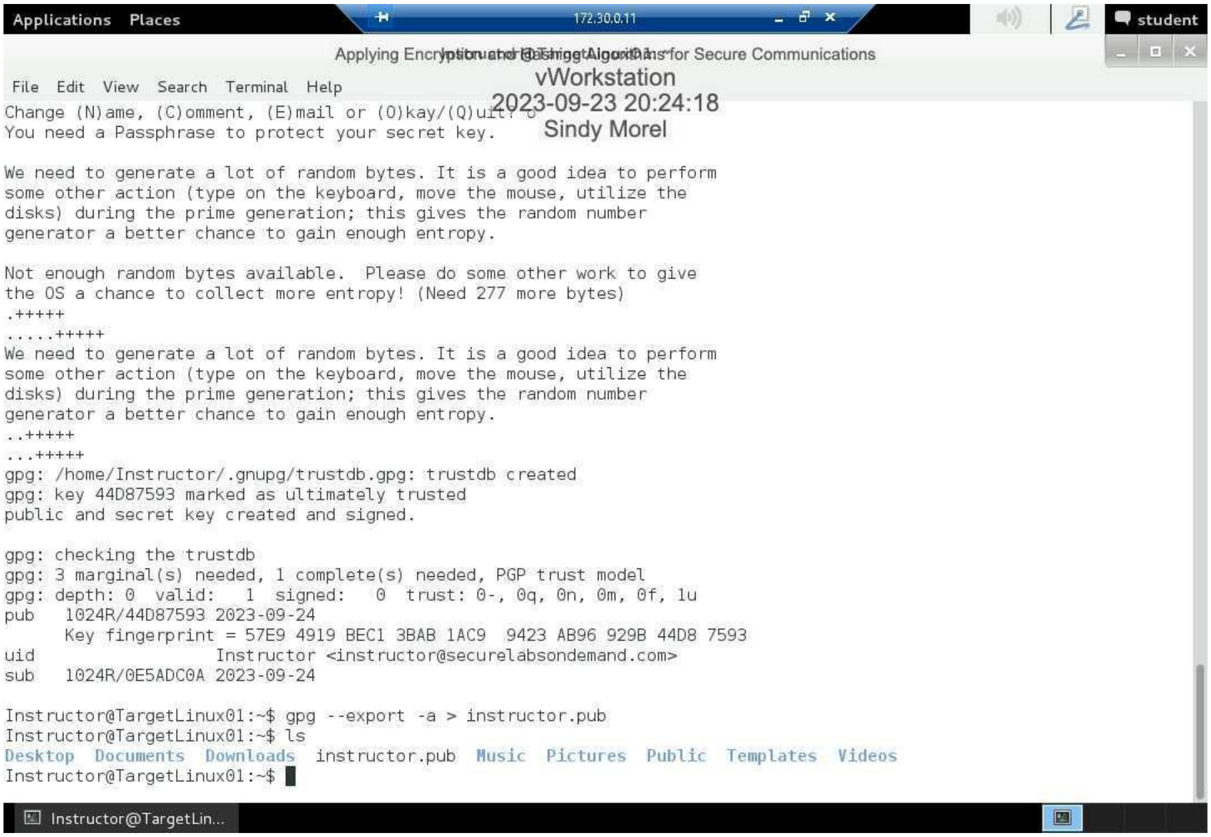


```
student@TargetLinux01:~$ cd Documents
student@TargetLinux01:~/Documents$ ls -l
total 4
-rw-r--r-- 1 student student 22 Sep 23 16:38 Send.txt
student@TargetLinux01:~/Documents$ cat Send.txt
My name is sindymorel
student@TargetLinux01:~/Documents$ md5sum Send.txt
5db8198da3043a36cbce8c6471c6fba8  Send.txt
student@TargetLinux01:~/Documents$ md5sum Send.txt > Send.txt.md5
student@TargetLinux01:~/Documents$ ls
Send.txt  Send.txt.md5
student@TargetLinux01:~/Documents$ cat Send.txt.md5
5db8198da3043a36cbce8c6471c6fba8  Send.txt
student@TargetLinux01:~/Documents$
```

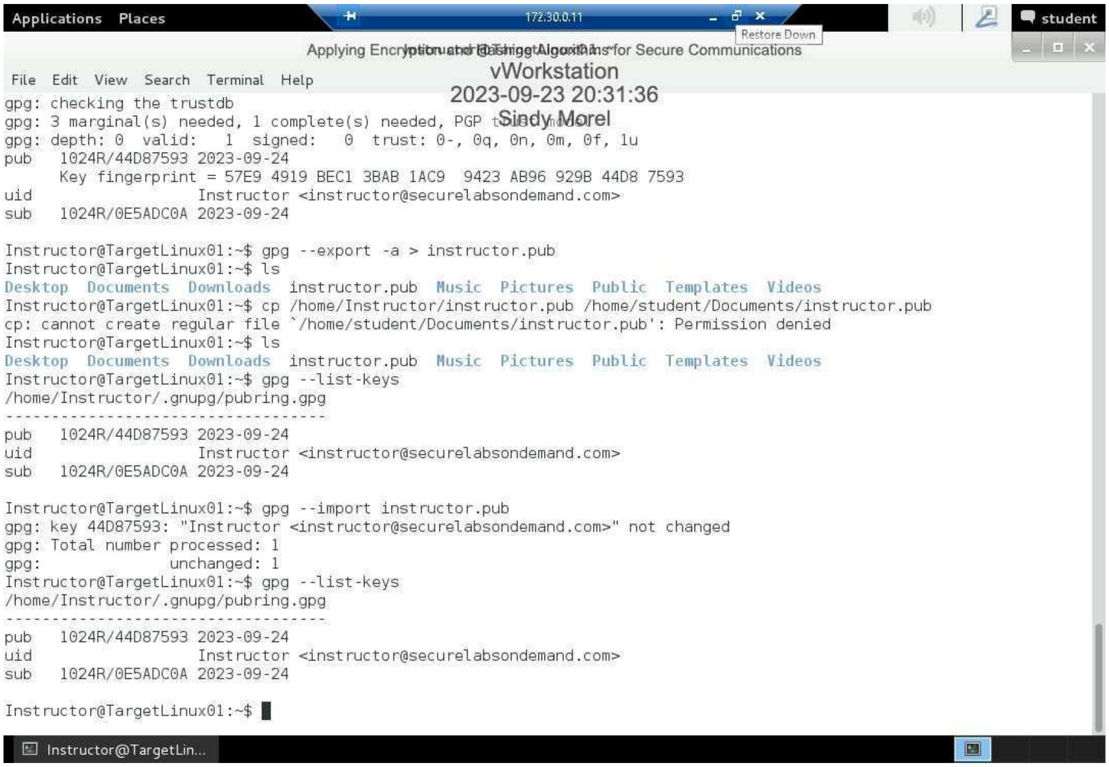

Applications
Places
172.30.0.11
student
Applying Encryption and Hashing Algorithms for Secure Communications
student@TargetLinux01: ~/Documents
File Edit View Search Terminal Help
student@TargetLinux...

vWorkstation
2023-09-23 20:07:34
Sindy Morel

```
generator a better chance to gain enough entropy.

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 283 more bytes)
.+++++
.+++++
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
...+++++
.+++++
gpg: /home/student/.gnupg/trustdb.gpg: trustdb created
gpg: key 603A4BEA marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   1024R/603A4BEA 2023-09-23
      Key fingerprint = 3B4A 3DA1 22A2 E1ED BCDD  2CE5 1758 5E4B 603A 4BEA
uid                  Student <student@securelabsondemand.com>
sub   1024R/FC85B90E 2023-09-23

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

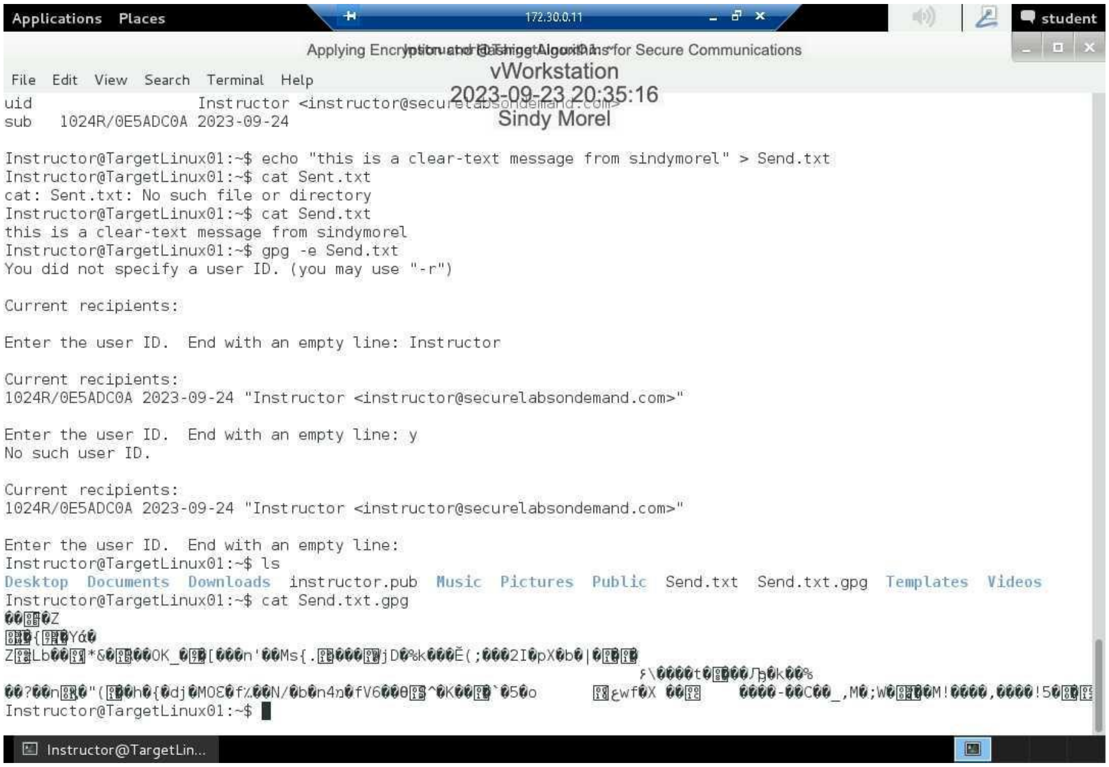


```
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 277 more bytes)
.+++++
.....+++++
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
..+++++
...+++++
gpg: /home/Instructor/.gnupg/trustdb.gpg: trustdb created
gpg: key 44D87593 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   1024R/44D87593 2023-09-24
      Key fingerprint = 57E9 4919 BEC1 3BAB 1AC9  9423 AB96 929B 44D8 7593
uid                  Instructor <instructor@securelabsondemand.com>
sub   1024R/0E5ADC0A 2023-09-24

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



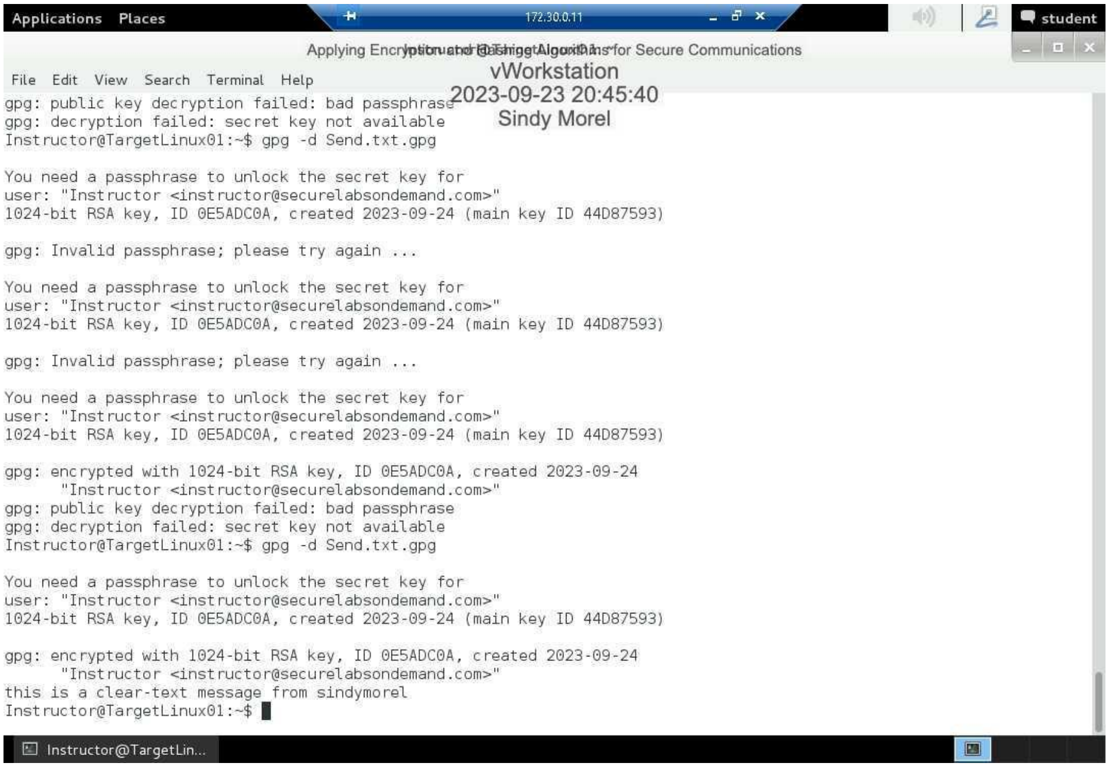
Applications Places
172.30.0.11
Restore Down
student
Applying Encryption and Hashing Algorithms for Secure Communications
Instructor@TargetLinux01:~
File Edit View Search Terminal Help
vWorkstation
2023-09-23 20:31:36
Sindy Morel
Instructor@TargetLin...

```
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   1024R/44D87593 2023-09-24
      Key fingerprint = 57E9 4919 BEC1 3BAB 1AC9  9423 AB96 929B 44D8 7593
uid                  Instructor <instructor@securelabsondemand.com>
sub   1024R/0E5ADC0A 2023-09-24

Instructor@TargetLinux01:~$ gpg --export -a > instructor.pub
Instructor@TargetLinux01:~$ ls
Desktop  Documents  Downloads  instructor.pub  Music  Pictures  Public  Templates  Videos
Instructor@TargetLinux01:~$ cp /home/Instructor/instructor.pub /home/student/Documents/instructor.pub
cp: cannot create regular file `/home/student/Documents/instructor.pub': Permission denied
Instructor@TargetLinux01:~$ ls
Desktop  Documents  Downloads  instructor.pub  Music  Pictures  Public  Templates  Videos
Instructor@TargetLinux01:~$ gpg --list-keys
/home/Instructor/.gnupg/pubring.gpg
-----------------------------------
pub   1024R/44D87593 2023-09-24
uid                  Instructor <instructor@securelabsondemand.com>
sub   1024R/0E5ADC0A 2023-09-24

Instructor@TargetLinux01:~$ gpg --import instructor.pub
gpg: key 44D87593: "Instructor <instructor@securelabsondemand.com>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
Instructor@TargetLinux01:~$ gpg --list-keys
/home/Instructor/.gnupg/pubring.gpg
-----------------------------------
pub   1024R/44D87593 2023-09-24
uid                  Instructor <instructor@securelabsondemand.com>
sub   1024R/0E5ADC0A 2023-09-24

Instructor@TargetLinux01:~$ █
```


Applications
Places
172.30.0.11
student
Applying Encryption and Hashing Algorithms for Secure Communications
Instructor@TargetLinux01:~
File
Edit
View
Search
Terminal
Help
Instructor@TargetLin...

vWorkstation
2023-09-23 20:35:16
Sindy Morel

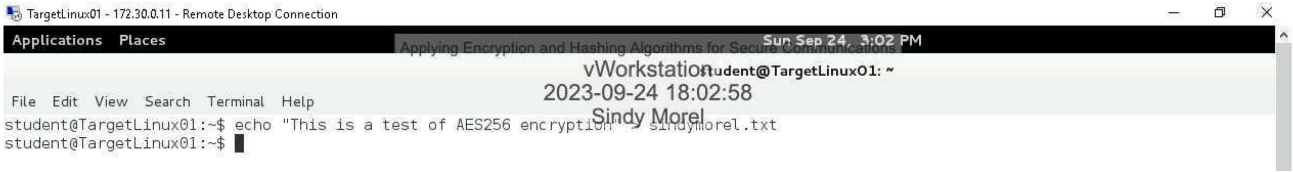
```
uid                  Instructor <instructor@securelabsondemand.com>
sub   1024R/0E5ADC0A 2023-09-24

Instructor@TargetLinux01:~$ echo "this is a clear-text message from sindymorel" > Send.txt
Instructor@TargetLinux01:~$ cat Sent.txt
cat: Sent.txt: No such file or directory
Instructor@TargetLinux01:~$ cat Send.txt
this is a clear-text message from sindymorel
Instructor@TargetLinux01:~$ 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: Instructor

Current recipients:
1024R/0E5ADC0A 2023-09-24 "Instructor <instructor@securelabsondemand.com>"

Enter the user ID.  End with an empty line: y
No such user ID.

Current recipients:
1024R/0E5ADC0A 2023-09-24 "Instructor <instructor@securelabsondemand.com>"

Enter the user ID.  End with an empty line:
Instructor@TargetLinux01:~$ ls
Desktop  Documents  Downloads  instructor.pub  Music  Pictures  Public  Send.txt  Send.txt.gpg  Templates  Videos
Instructor@TargetLinux01:~$ cat Send.txt.gpg
��[illegible]�Z
[illegible]�{[illegible]�Yά�
Z[illegible]Lb��[illegible]*&�[illegible]���0K_�[illegible][���n'��Ms{.[illegible]���[illegible]jD�%k���Ĕ(;���2I�pX�b�|�[illegible]
                                                                                                    ۶\����t�[illegible]��Л�k��%

��?��n[illegible]�"([illegible]�h�{�dj�M0Ɛ�f٪��N/�b�n4ɲ�fV6���[illegible]^�K��[illegible]`�5�o      [illegible]ɛwf�X ��[illegible]    ����-��C��_,M�;W�[illegible]�M!����,����!5�[illegible]
Instructor@TargetLinux01:~$ 
```



Applications Places
172.30.0.11
student
Applying Encryption and Hashing Algorithms for Secure Communications
Instructor@TargetLinux01: ~
vWorkstation
2023-09-23 20:45:40
Sindy Morel
File Edit View Search Terminal Help
Instructor@TargetLin...

```
gpg: public key decryption failed: bad passphrase
gpg: decryption failed: secret key not available
Instructor@TargetLinux01:~$ gpg -d Send.txt.gpg

You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 0E5ADC0A, created 2023-09-24 (main key ID 44D87593)

gpg: Invalid passphrase; please try again ...

You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 0E5ADC0A, created 2023-09-24 (main key ID 44D87593)

gpg: Invalid passphrase; please try again ...

You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 0E5ADC0A, created 2023-09-24 (main key ID 44D87593)

gpg: encrypted with 1024-bit RSA key, ID 0E5ADC0A, created 2023-09-24
      "Instructor <instructor@securelabsondemand.com>"
gpg: public key decryption failed: bad passphrase
gpg: decryption failed: secret key not available
Instructor@TargetLinux01:~$ gpg -d Send.txt.gpg

You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 0E5ADC0A, created 2023-09-24 (main key ID 44D87593)

gpg: encrypted with 1024-bit RSA key, ID 0E5ADC0A, created 2023-09-24
      "Instructor <instructor@securelabsondemand.com>"
this is a clear-text message from sindymorel
Instructor@TargetLinux01:~$
```

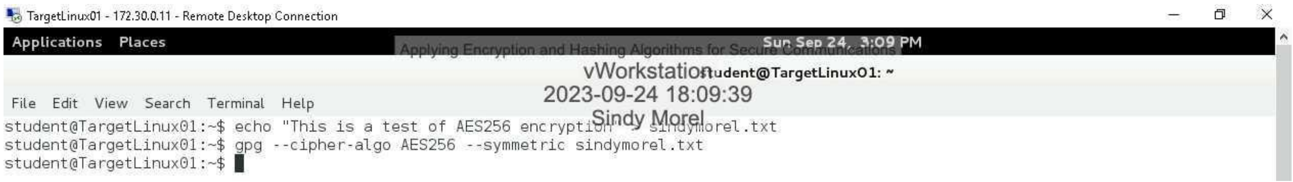
Part 3: Challenge Exercise

Create a new file called sindymorel.txt


TargetLinux01 - 172.30.0.11 - Remote Desktop Connection
Applications Places
Applying Encryption and Hashing Algorithms for Secure Communications
Sun Sep 24, 3:02 PM
student@TargetLinux01: ~
vWorkstation
2023-09-24 18:02:58
Sindy Morel
File Edit View Search Terminal Help

```
student@TargetLinux01:~$ echo "This is a test of AES256 encryption" > sindymorel.txt
student@TargetLinux01:~$
```

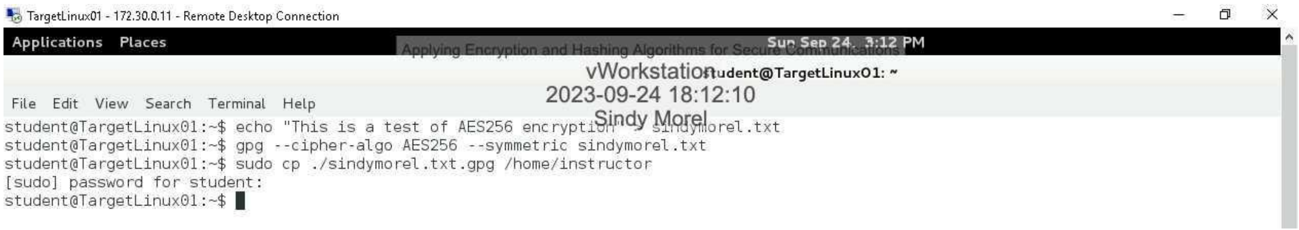
Encrypting File


TargetLinux01 - 172.30.0.11 - Remote Desktop Connection
Applications Places
Applying Encryption and Hashing Algorithms for Secure Communications
Sun Sep 24, 3:09 PM
student@TargetLinux01: ~
vWorkstation
2023-09-24 18:09:39
Sindy Morel
File Edit View Search Terminal Help

```
student@TargetLinux01:~$ echo "This is a test of AES256 encryption" > sindymorel.txt
student@TargetLinux01:~$ gpg --cipher-algo AES256 --symmetric sindymorel.txt
student@TargetLinux01:~$
```

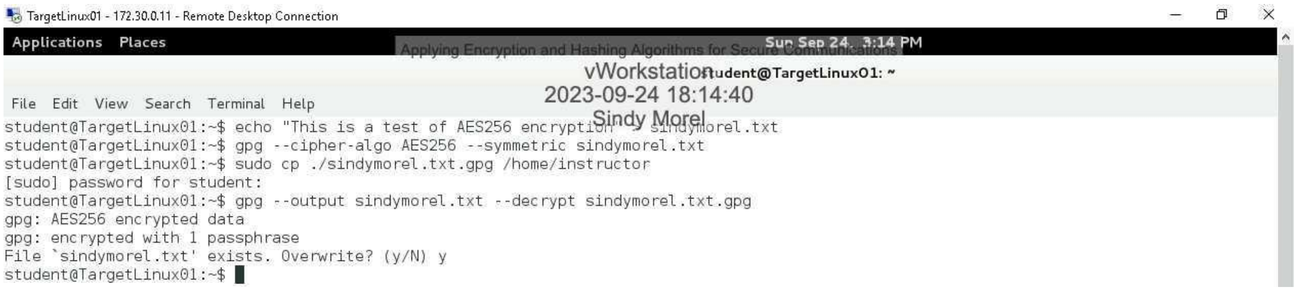
Sindy Morel	LAB7: Applying Encryption and Hashing Algorithms for Secure Communications	September 23, 2023

Copying File

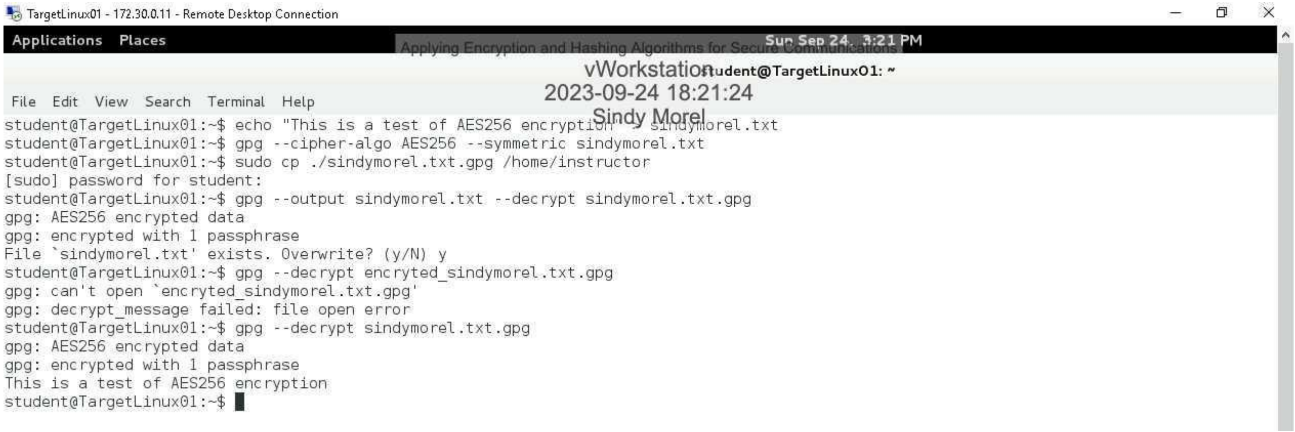


```
student@TargetLinux01:~$ echo "This is a test of AES256 encryption" > sindymorel.txt
student@TargetLinux01:~$ gpg --cipher-algo AES256 --symmetric sindymorel.txt
student@TargetLinux01:~$ sudo cp ./sindymorel.txt.gpg /home/instructor
[sudo] password for student:
student@TargetLinux01:~$
```

Decrypting File



```
student@TargetLinux01:~$ echo "This is a test of AES256 encryption" > sindymorel.txt
student@TargetLinux01:~$ gpg --cipher-algo AES256 --symmetric sindymorel.txt
student@TargetLinux01:~$ sudo cp ./sindymorel.txt.gpg /home/instructor
[sudo] password for student:
student@TargetLinux01:~$ gpg --output sindymorel.txt --decrypt sindymorel.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
File `sindymorel.txt' exists. Overwrite? (y/N) y
student@TargetLinux01:~$
```



```
student@TargetLinux01:~$ echo "This is a test of AES256 encryption" > sindymorel.txt
student@TargetLinux01:~$ gpg --cipher-algo AES256 --symmetric sindymorel.txt
student@TargetLinux01:~$ sudo cp ./sindymorel.txt.gpg /home/instructor
[sudo] password for student:
student@TargetLinux01:~$ gpg --output sindymorel.txt --decrypt sindymorel.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
File `sindymorel.txt' exists. Overwrite? (y/N) y
student@TargetLinux01:~$ gpg --decrypt encryted_sindymorel.txt.gpg
gpg: can't open `encryted_sindymorel.txt.gpg'
gpg: decrypt_message failed: file open error
student@TargetLinux01:~$ gpg --decrypt sindymorel.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
This is a test of AES256 encryption
student@TargetLinux01:~$
```