

Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

Student:

Patrick Kierzkowski

Email:

pxk405@francis.edu

Time on Task:

10 hours, 0 minutes

Progress:

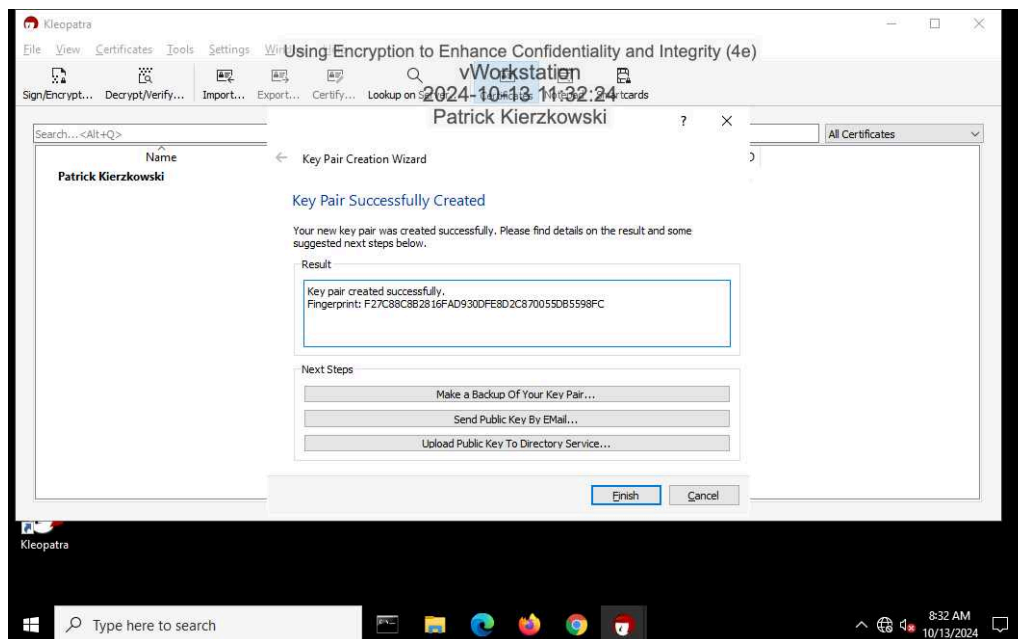
100%

Report Generated: Monday, July 7, 2025 at 9:50 PM

Section 1: Hands-On Demonstration

Part 1: Create and Exchange Asymmetric Encryption Keys

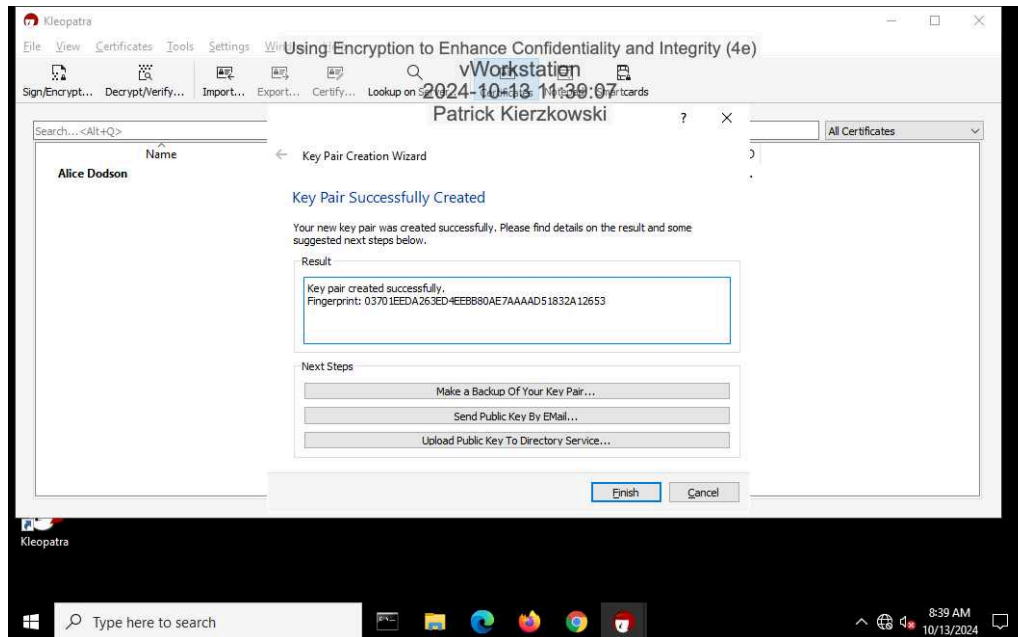
9. Make a screen capture showing the **fingerprint** for your key pair.



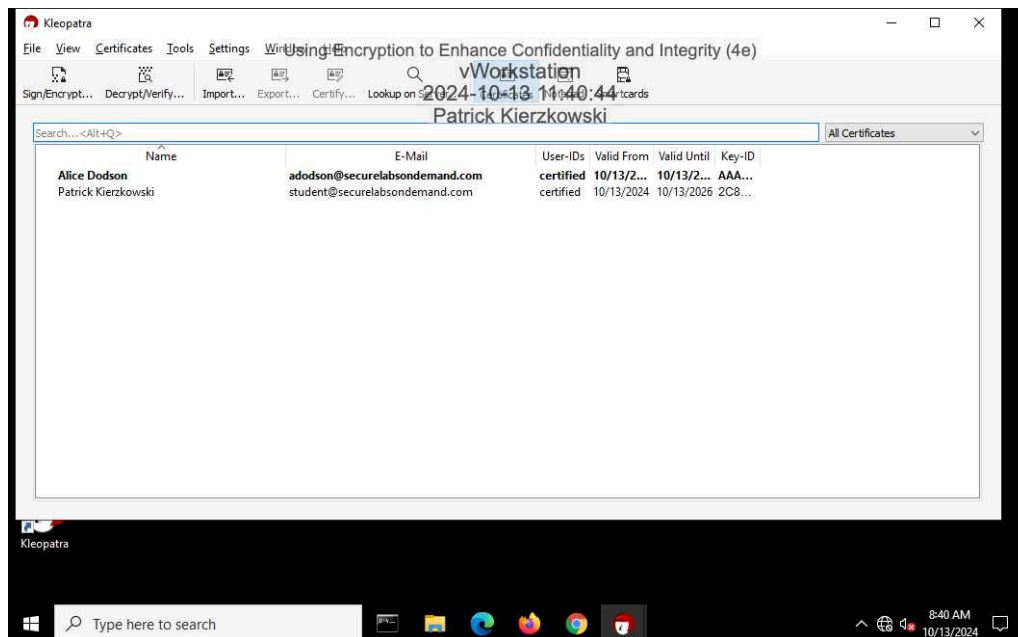
Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

22. Make a screen capture showing the **fingerprint** for Alice's key pair.



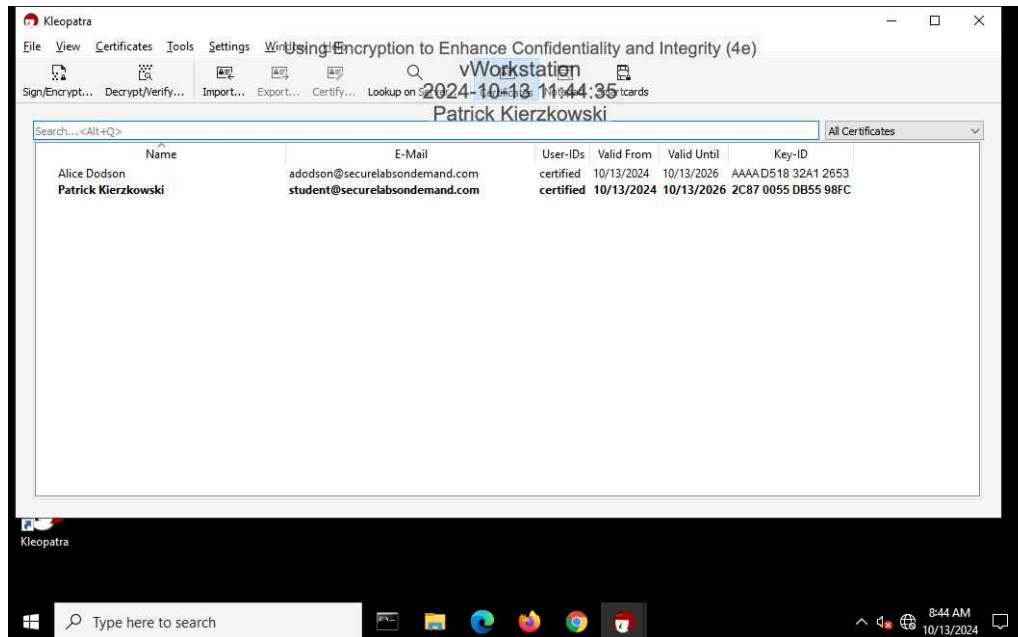
30. Make a screen capture showing your public key in Alice's certificate cache.



Using Encryption to Enhance Confidentiality and Integrity (4e)

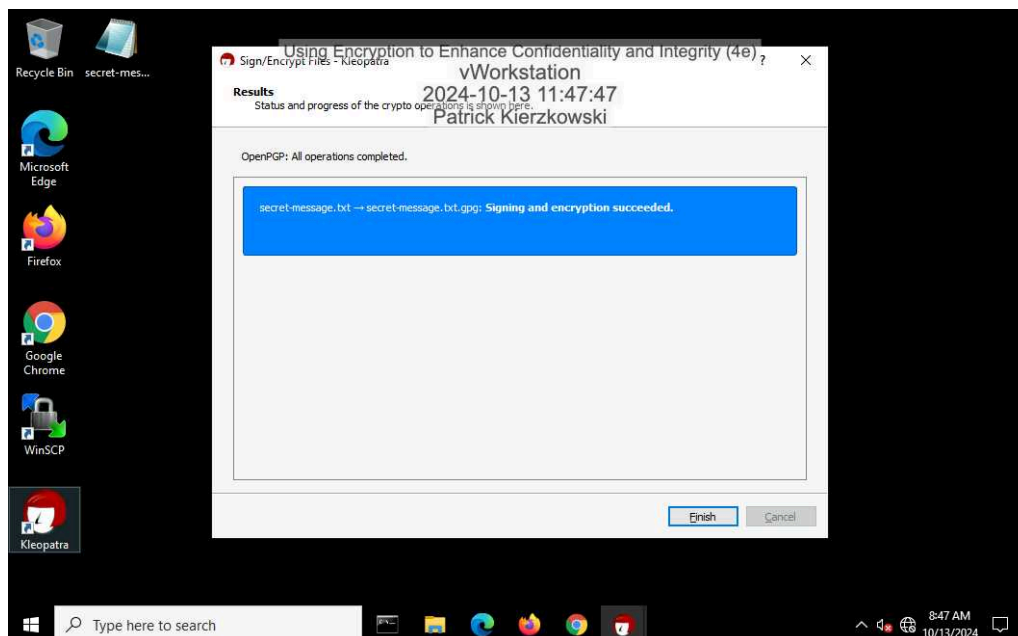
Fundamentals of Information Systems Security, Fourth Edition - Lab 05

35. Make a screen capture showing Alice's public key in your certificate cache.



Part 2: Encrypt a File Using Asymmetric Encryption

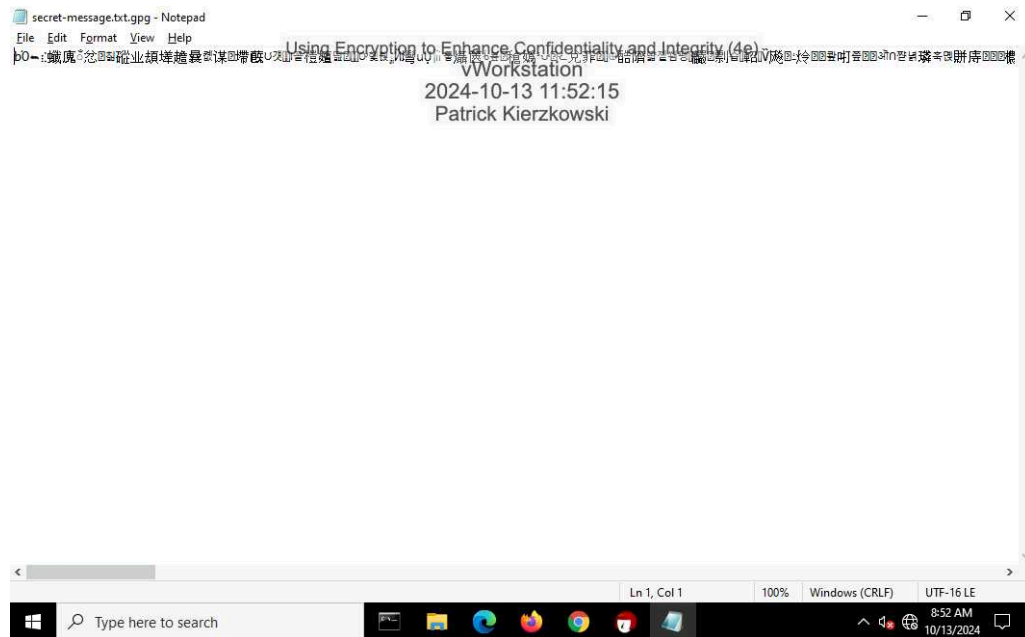
9. Make a screen capture showing the successful signing and encryption message.



Using Encryption to Enhance Confidentiality and Integrity (4e)

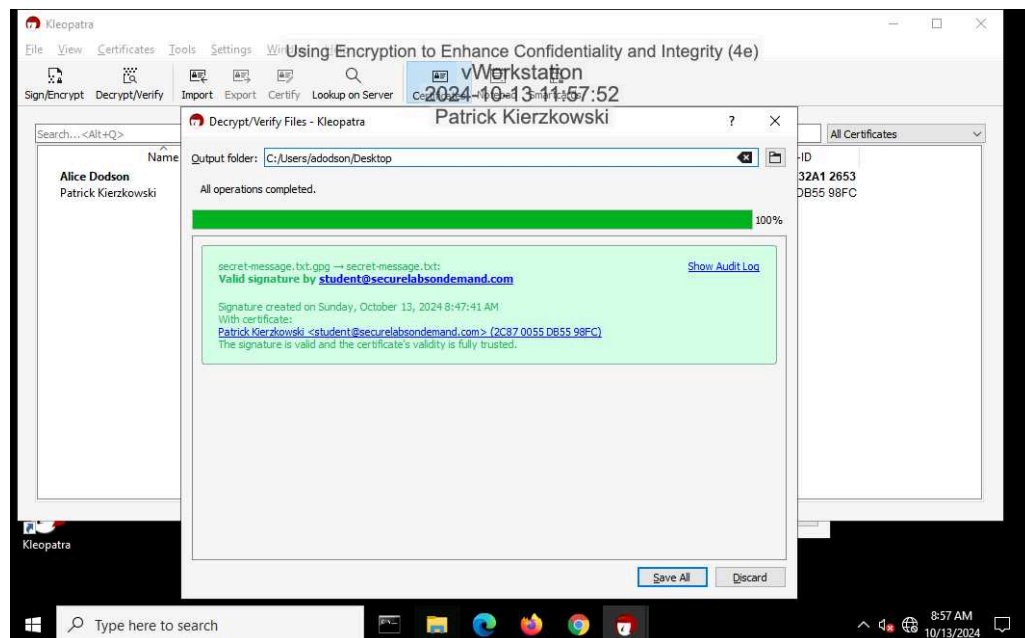
Fundamentals of Information Systems Security, Fourth Edition - Lab 05

12. Make a screen capture showing the **ciphertext**.



Part 3: Decrypt a File Using Asymmetric Encryption

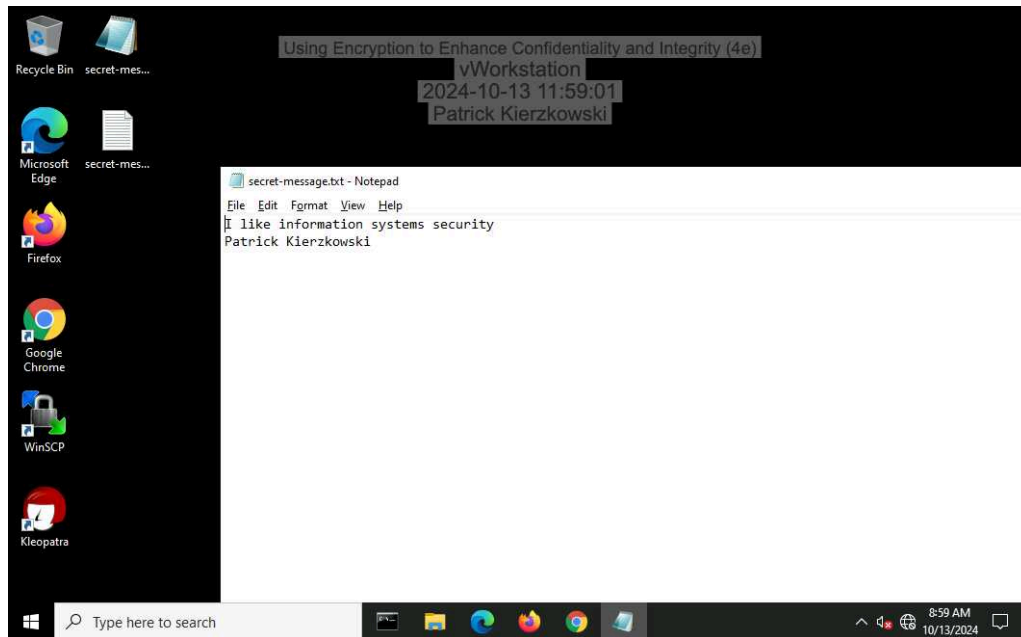
15. Make a screen capture showing the **Decrypt/Verify Files** window.



Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

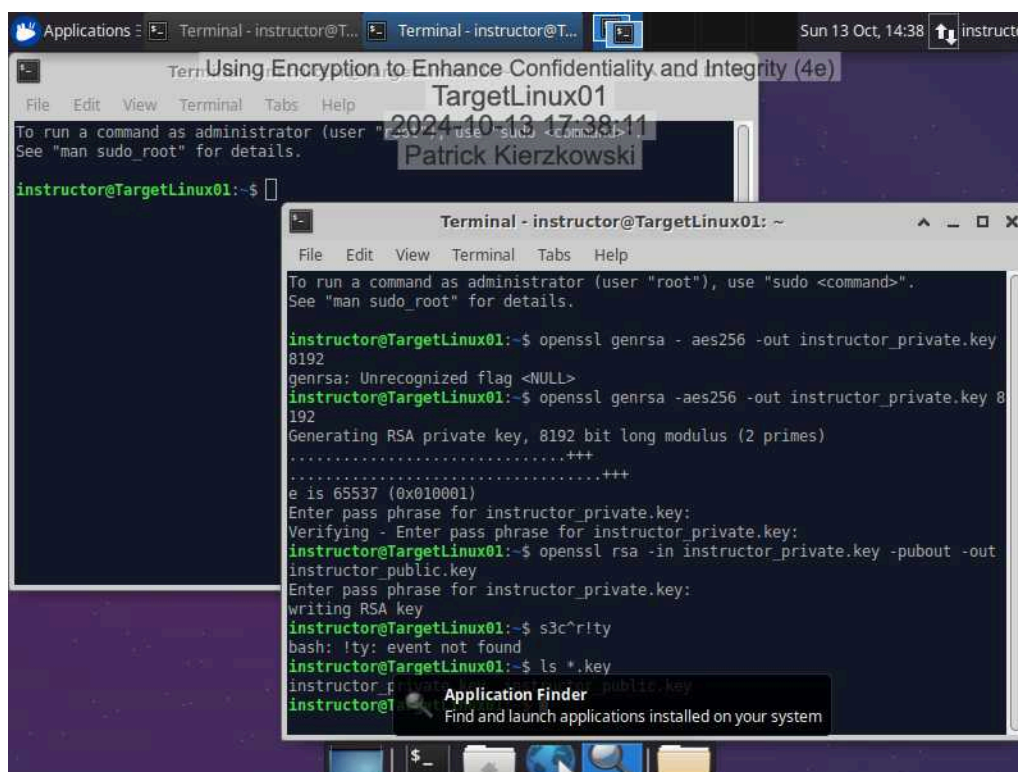
18. Make a screen capture showing the **decrypted secret-message.txt** file in Notepad.



Section 2: Applied Learning

Part 1: Create an Asymmetric Key Pair

10. Make a screen capture showing the instructor's key pair files.



Part 2: Encrypt a File Using Symmetric Encryption

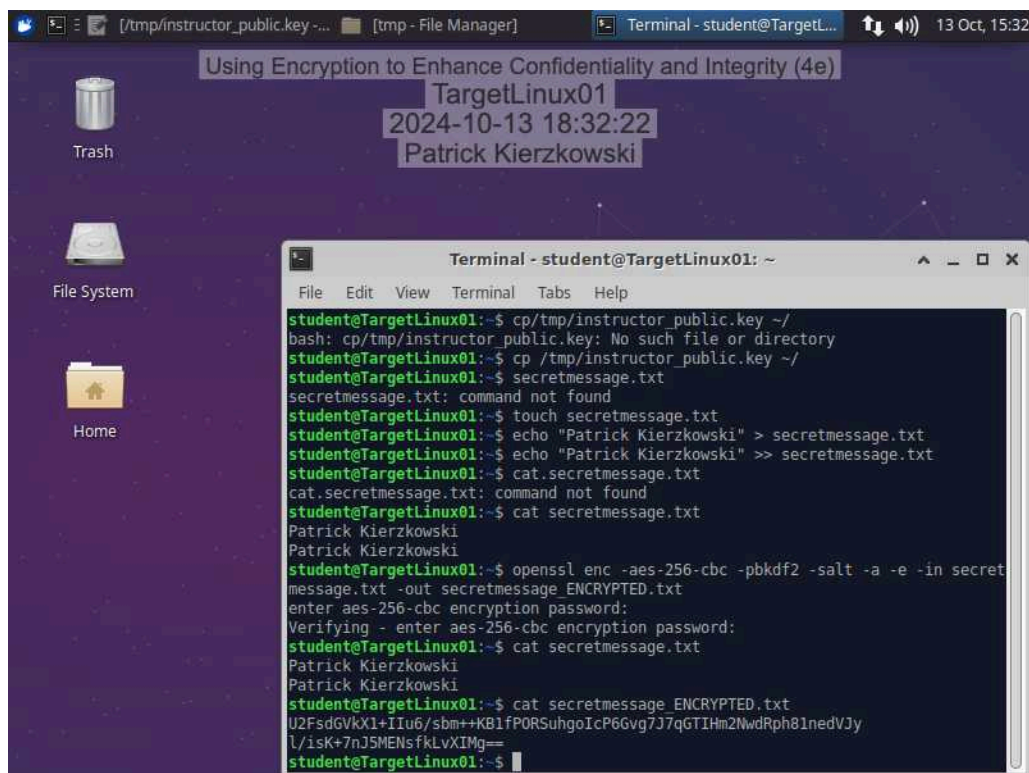
11. Document the password you used to symmetrically encrypt the file.

yourownpassword

Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

13. Make a screen capture showing the ciphertext in the `secretmessage_ENCRYPTED.txt` file.



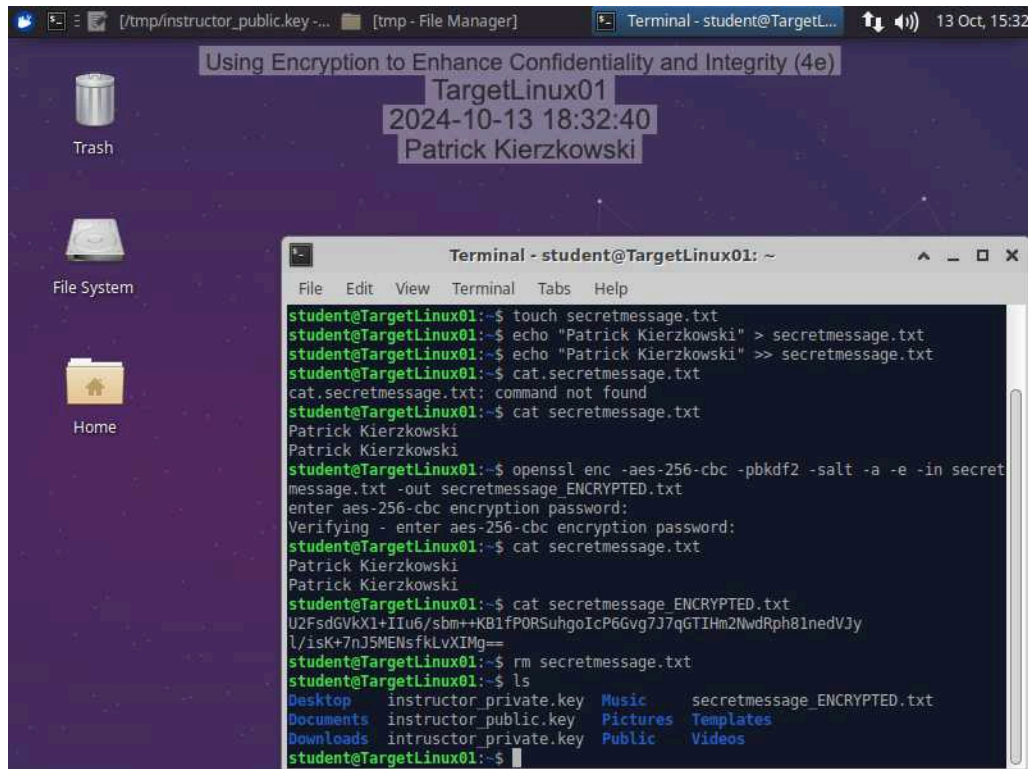
The screenshot shows a Linux desktop with a purple background. A terminal window titled "Terminal - student@TargetLinux01" is open, displaying the following commands and output:

```
student@TargetLinux01:~$ cp/tmp/instructor_public.key ~/
bash: cp/tmp/instructor_public.key: No such file or directory
student@TargetLinux01:~$ cp /tmp/instructor_public.key ~/
student@TargetLinux01:~$ secretmessage.txt
secretmessage.txt: command not found
student@TargetLinux01:~$ touch secretmessage.txt
student@TargetLinux01:~$ echo "Patrick Kierzkowski" > secretmessage.txt
student@TargetLinux01:~$ echo "Patrick Kierzkowski" >> secretmessage.txt
student@TargetLinux01:~$ cat.secretmessage.txt
cat.secretmessage.txt: command not found
student@TargetLinux01:~$ cat secretmessage.txt
Patrick Kierzkowski
Patrick Kierzkowski
student@TargetLinux01:~$ openssl enc -aes-256-cbc -pbkdf2 -salt -a -e -in secret
message.txt -out secretmessage_ENCRYPTED.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
student@TargetLinux01:~$ cat secretmessage.txt
Patrick Kierzkowski
Patrick Kierzkowski
student@TargetLinux01:~$ cat secretmessage_ENCRYPTED.txt
U2FsdkGVkX1+IIu6/sbm++KB1fPORSuhgoIcP6Gvg7J7qGTIHm2NwdRph81nedVJy
l/1sK+7nJ5MENsfkLvXIMg==
student@TargetLinux01:~$
```


Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

16. Make a screen capture showing the output of the **ls** command.



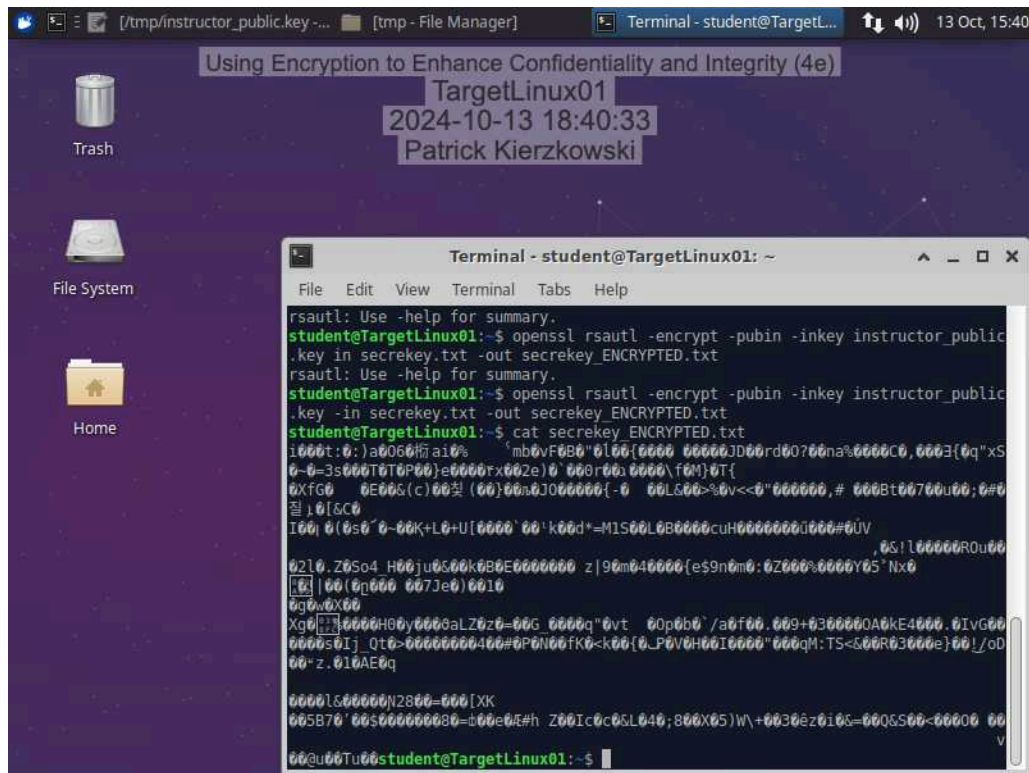
```
student@TargetLinux01:~$ touch secretmessage.txt
student@TargetLinux01:~$ echo "Patrick Kierzkowski" > secretmessage.txt
student@TargetLinux01:~$ echo "Patrick Kierzkowski" >> secretmessage.txt
student@TargetLinux01:~$ cat secretmessage.txt
cat: secretmessage.txt: command not found
student@TargetLinux01:~$ cat secretmessage.txt
Patrick Kierzkowski
Patrick Kierzkowski
student@TargetLinux01:~$ openssl enc -aes-256-cbc -pbkdf2 -salt -a -e -in secretmessage.txt -out secretmessage_ENCRYPTED.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
student@TargetLinux01:~$ cat secretmessage.txt
Patrick Kierzkowski
Patrick Kierzkowski
student@TargetLinux01:~$ cat secretmessage_ENCRYPTED.txt
U2FsdGVkX1+IIu6/sbm++KB1fPORSuhgoIcP6Gvg7J7qGTIHm2NwdRph81nedVJy
l/isK+7nJ5MENsfkLvXIMg==
student@TargetLinux01:~$ rm secretmessage.txt
student@TargetLinux01:~$ ls
Desktop  instructor_private.key  Music      secretmessage_ENCRYPTED.txt
Documents  instructor_public.key  Pictures   Templates
Downloads  instructor_private.key  Public     Videos
student@TargetLinux01:~$
```

Part 3: Transfer and Decrypt a File Using Hybrid Cryptography

Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

6. Make a screen capture showing the encrypted contents of the `secretkey_ENCRYPTED.txt` file.



The screenshot shows a Linux desktop with a purple background. A terminal window titled "Terminal - student@TargetLinux01" is open, displaying the following commands and output:

```
rsautl: Use -help for summary.
student@TargetLinux01:~$ openssl rsautl -encrypt -pubin -inkey instructor_public
.key in secrekey.txt -out secrekey_ENCRYPTED.txt
rsautl: Use -help for summary.
student@TargetLinux01:~$ openssl rsautl -encrypt -pubin -inkey instructor_public
.key -in secrekey.txt -out secrekey_ENCRYPTED.txt
student@TargetLinux01:~$ cat secrekey_ENCRYPTED.txt
```

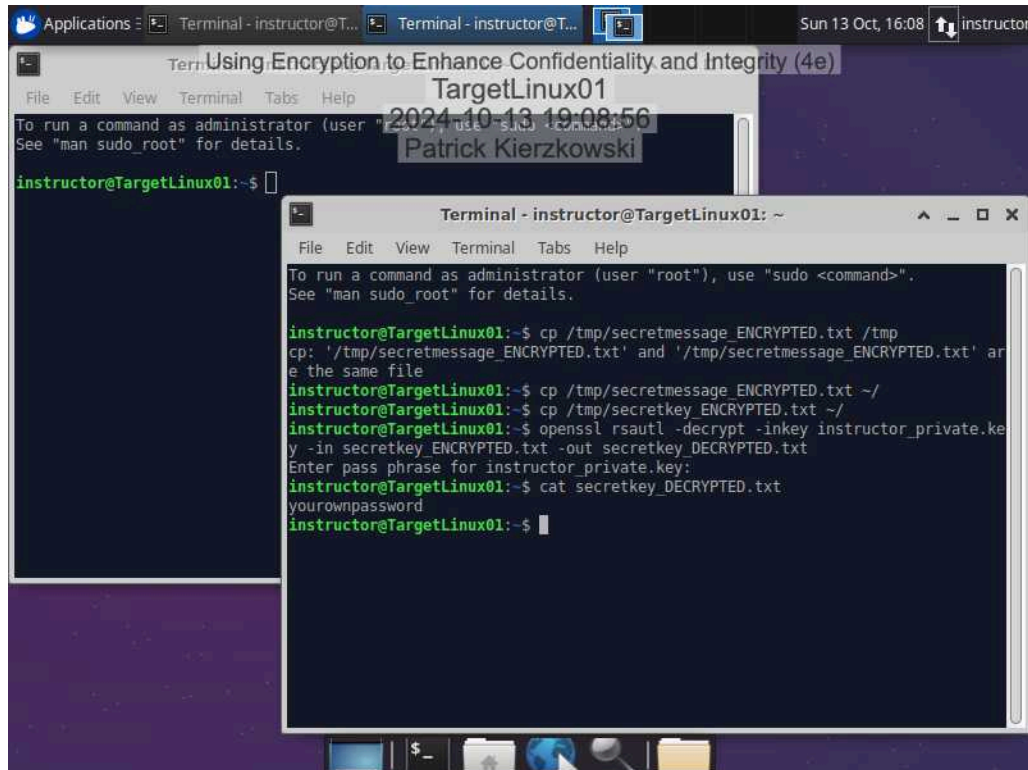
The output of the `cat` command shows the encrypted contents of the file, which is a base64-encoded string:

```
i000t:0:)a0060f ai0% 'mb0vF0B0"0l00{0000 00000JD00rd00?00na%0000C0,0003{0q"xS
0-0=3s000T0T0P00)e0000rx002e)0'000-00:0000\fm)0T{
0XfG0 0E00G(c)003! (00)00n0J000000{-0 00L600>:0v<0"000000,# 000Bt00700u00;0#0
00 10[&c0
I00i 0(0s0"0-00K+L0+U{0000'00'k00d*=MIS00L0B0000cuH00000000000#0UV
,0S!l00000R0u00
02l0.Z0So4 H00ju0500k0B0E0000000 z|90m040000{eS9n0m0:0Z000?0000Y05"Nx0
00|00(0p000 007Je0)0010
0g0v0X00
Xg0[0000H00y0000aL0z0=00G 0000q"0vt 00p0b0'/a0f00.009+0300000A0kE4000.0IvG00
0000s0Ij Qt0>00000000400#0P0N00fK0<k00{0_P0V0H00I0000"000qM:TS<S00R03000e}00!/oD
00"z.010AE0q
0000L&00000jN2800=000[XK
00SB70'00S000000000=000e04#h Z00Ic0c0SL040;800X05)W\+0030Ez0i0S=00QSS00<00000 00
v
00@u00Tu00student@TargetLinux01:~$
```

Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

17. Make a screen capture showing the **decrypted contents of the secretkey_DECRYPTED.txt file**.



The screenshot shows a terminal window titled "Terminal - instructor@TargetLinux01: ~". The window displays the following commands and output:

```
instructor@TargetLinux01:~$ cp /tmp/secretmessage_ENCRYPTED.txt /tmp
cp: '/tmp/secretmessage_ENCRYPTED.txt' and '/tmp/secretmessage_ENCRYPTED.txt' are the same file
instructor@TargetLinux01:~$ cp /tmp/secretmessage_ENCRYPTED.txt ~/
instructor@TargetLinux01:~$ cp /tmp/secretkey_ENCRYPTED.txt ~/
instructor@TargetLinux01:~$ openssl rsautl -decrypt -inkey instructor_private.key -in secretkey_ENCRYPTED.txt -out secretkey_DECRYPTED.txt
Enter pass phrase for instructor_private.key:
instructor@TargetLinux01:~$ cat secretkey_DECRYPTED.txt
yourownpassword
instructor@TargetLinux01:~$
```

The terminal window also shows a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The system clock in the top right corner indicates "Sun 13 Oct, 16:08".

Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

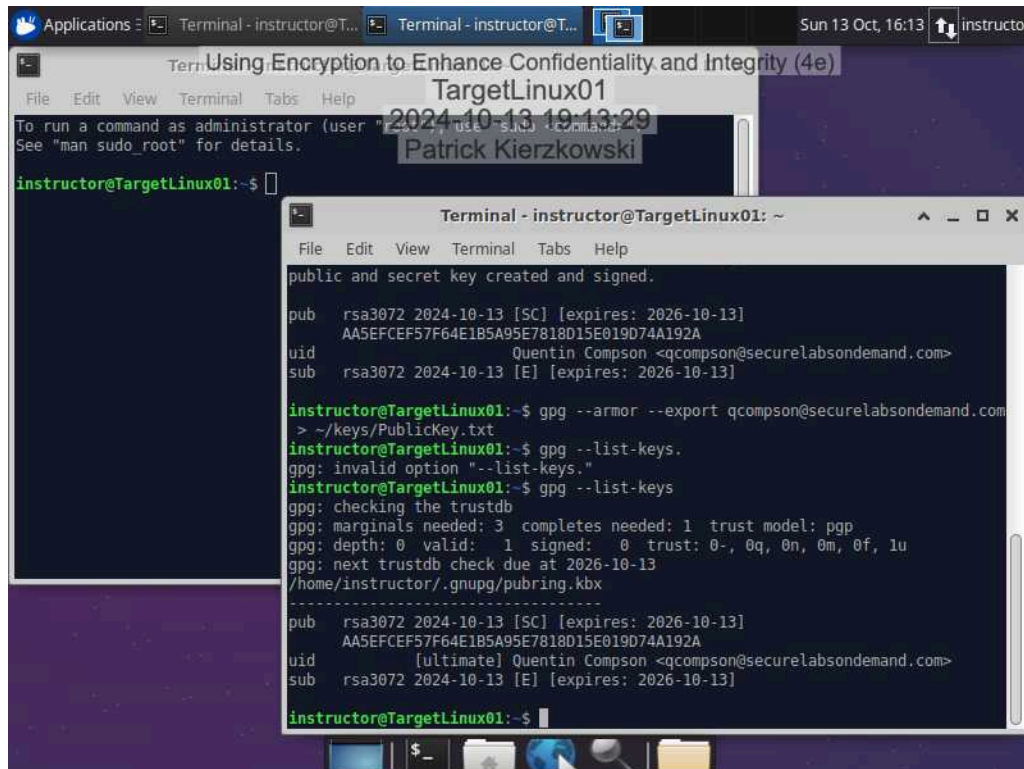
21. Make a screen capture showing the contents of the `secretmessage_DECRYPTED` file.

```
instructor@TargetLinux01:~$ cp /tmp/secretmessage_ENCRYPTED.txt /tmp
cp: '/tmp/secretmessage_ENCRYPTED.txt' and '/tmp/secretmessage_ENCRYPTED.txt' are the same file
instructor@TargetLinux01:~$ cp /tmp/secretmessage_ENCRYPTED.txt ~/
instructor@TargetLinux01:~$ cp /tmp/secretkey_ENCRYPTED.txt ~/
instructor@TargetLinux01:~$ openssl rsautl -decrypt -inkey instructor_private.key -in secretkey_ENCRYPTED.txt -out secretkey_DECRYPTED.txt
Enter pass phrase for instructor_private.key:
yourpassword
instructor@TargetLinux01:~$ cat secretkey_DECRYPTED.txt
yourpassword
instructor@TargetLinux01:~$ openssl enc -aes-256-cbc -pbkdf2 -salt -a -d -in secretmessage_ENCRYPTED.txt -out secretmessage_DECRYPTED.txt
enter aes-256-cbc decryption password:
instructor@TargetLinux01:~$ cat secretmessage_DECRYPTED.txt
Patrick Kierzkowski
Patrick Kierzkowski
instructor@TargetLinux01:~$
```

Section 3: Challenge and Analysis

Part 1: Digitally Sign a Document Using GPG

Make a screen capture showing the **key fingerprint** for the key pair you generated in this part of the lab.



The screenshot shows a terminal window on a Linux system. The terminal title is "Terminal - instructor@TargetLinux01: ~". The prompt is "instructor@TargetLinux01:~\$". The user has run the command "gpg --full-gen-key", which has generated a new key pair. The output shows the key details: "pub rsa3072 2024-10-13 [SC] [expires: 2026-10-13] AA5EFCE57F64E1B5A95E7818D15E019D74A192A" and "uid [ultimate] Quentin Compson <qcompson@securelabsondemand.com>". The user has then run "gpg --armor --export qcompson@securelabsondemand.com" to export the public key to a file named "PublicKey.txt" in the directory "~/keys/". Finally, the user has run "gpg --list-keys", which displays the key details again, including the key fingerprint "AA5EFCE57F64E1B5A95E7818D15E019D74A192A".

```
instructor@TargetLinux01:~$ gpg --full-gen-key
gpg: keybox '/home/instructor/.gnupg/pubring.kbx'
gpg: public and secret key created and signed.

pub  rsa3072 2024-10-13 [SC] [expires: 2026-10-13]
     AA5EFCE57F64E1B5A95E7818D15E019D74A192A
uid   [ultimate] Quentin Compson <qcompson@securelabsondemand.com>
sub   rsa3072 2024-10-13 [E] [expires: 2026-10-13]

instructor@TargetLinux01:~$ gpg --armor --export qcompson@securelabsondemand.com
> ~/keys/PublicKey.txt
instructor@TargetLinux01:~$ gpg --list-keys
gpg: invalid option "--list-keys."
instructor@TargetLinux01:~$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid: 1  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2026-10-13
/home/instructor/.gnupg/pubring.kbx
-----
pub  rsa3072 2024-10-13 [SC] [expires: 2026-10-13]
     AA5EFCE57F64E1B5A95E7818D15E019D74A192A
uid   [ultimate] Quentin Compson <qcompson@securelabsondemand.com>
sub   rsa3072 2024-10-13 [E] [expires: 2026-10-13]

instructor@TargetLinux01:~$
```

Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

Make a screen capture showing the contents of the unsignedmessage.txt file.

```
instructor@TargetLinux01:~$ touch unsignedmessage.txt
instructor@TargetLinux01:~$ echo "This is a test message that will be digitally
signed by my own user account" unsignedmessage.txt
This is a test message that will be digitally signed by my own user account unsi
gnedmessage.txt
instructor@TargetLinux01:~$ echp "This is a test message that will be digitally
signed by my own user account" >> unsignedmessage.txt
Command 'echp' not found, did you mean:
  command 'ecp' from deb ecere-dev (0.44.15-1build3)
  command 'echo' from deb coreutils (8.30-3ubuntu2)
Try: sudo apt install <deb name>
instructor@TargetLinux01:~$ echo "This is a test message that will be digitally
signed by my own user account" >> unsignedmessage.txt
instructor@TargetLinux01:~$ cat unsignedmessage.txt
This is a test message that will be digitally signed by my own user account
instructor@TargetLinux01:~$
```

Part 2: Verify the Digital Signature Using Kleopatra

Using Encryption to Enhance Confidentiality and Integrity (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 05

Make a screen capture showing the **successful signature verification** on the signed message file.

