lab report:3

Course Name: Cyber Security and Digital Forensic

Course: CSE 414

Submitted by

Name: Shamiul Islam

Id:18192103241

Intake:41

Section:06

Program: B.Sc Eng. in CSE(BUBT)

Submitted To
Course Teacher:

Dr. Shekh Abdullah-Al-Musa Ahmed

lecturer

Department of CSE, BUBT

Submission Date:12-09-2022

Name of the experiment: RSA key encryption.

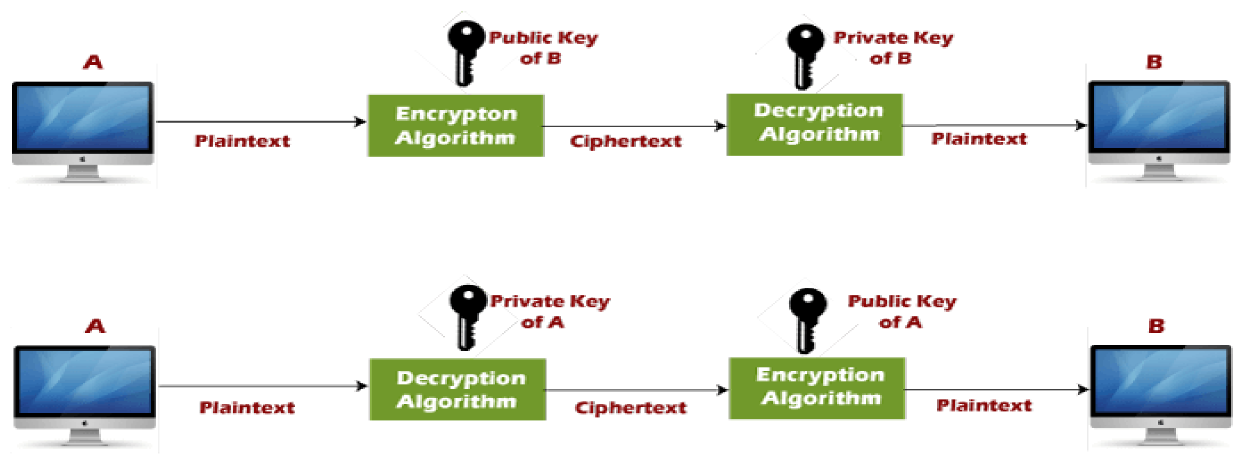
RSA: RSA(Rivest-Shamir-Adleman) is an Asymmetric encryption technique that uses two different keys as public and private keys to perform the encryption and decryption. With RSA, you can encrypt sensitive information with a public key and a matching private key is used to decrypt the encrypted message. Asymmetric encryption is mostly used when there are 2 different endpoints are involved such as VPN client and server, SSH, etc.

Public key encryption algorithm: Public Key encryption algorithm is also called the Asymmetric algorithm. Asymmetric algorithms are those algorithms in which sender and receiver use different keys for encryption and decryption. Each sender is assigned a pair of keys:

- Public key
- Private key

The Public key is used for encryption, and the Private Key is used for decryption. Decryption cannot be done using a public key. The two keys are linked, but the private key cannot be derived from the public key. The public key is well known, but the private key is secret and it is known only to the user who owns the key. It means that everybody can send a message to the user using user's public key. But only the user can decrypt the message using his private key.

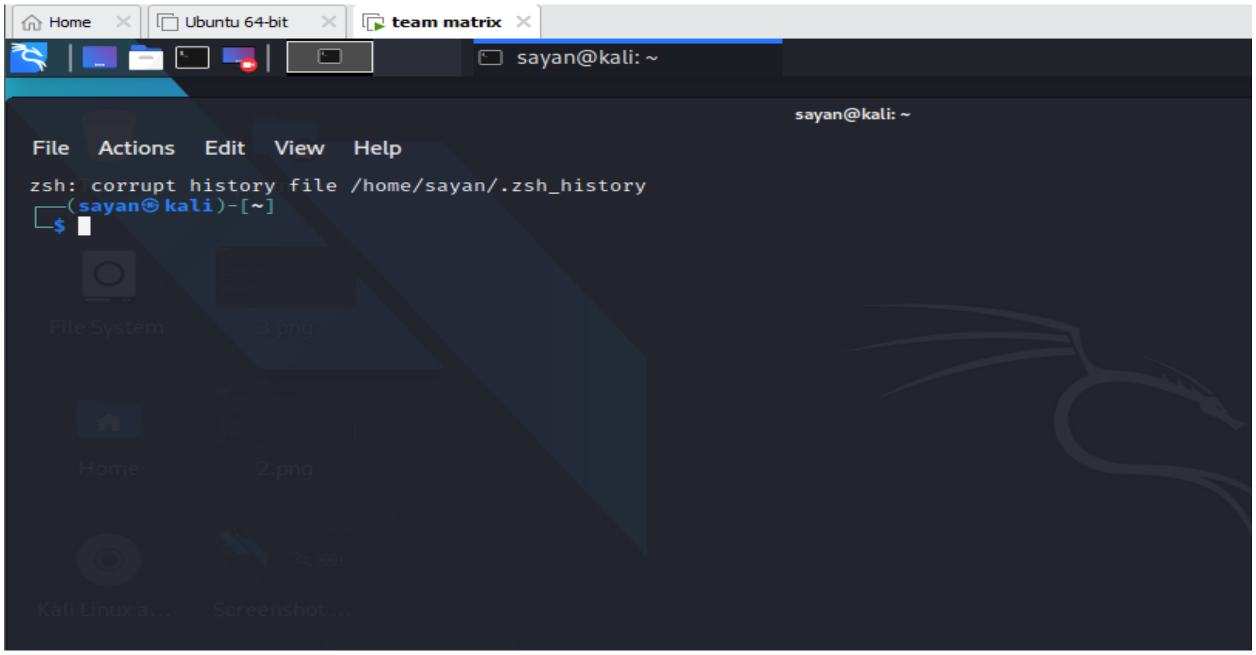
The Public key algorithm operates in the following manner:



Encryption/decryption using public/private keys

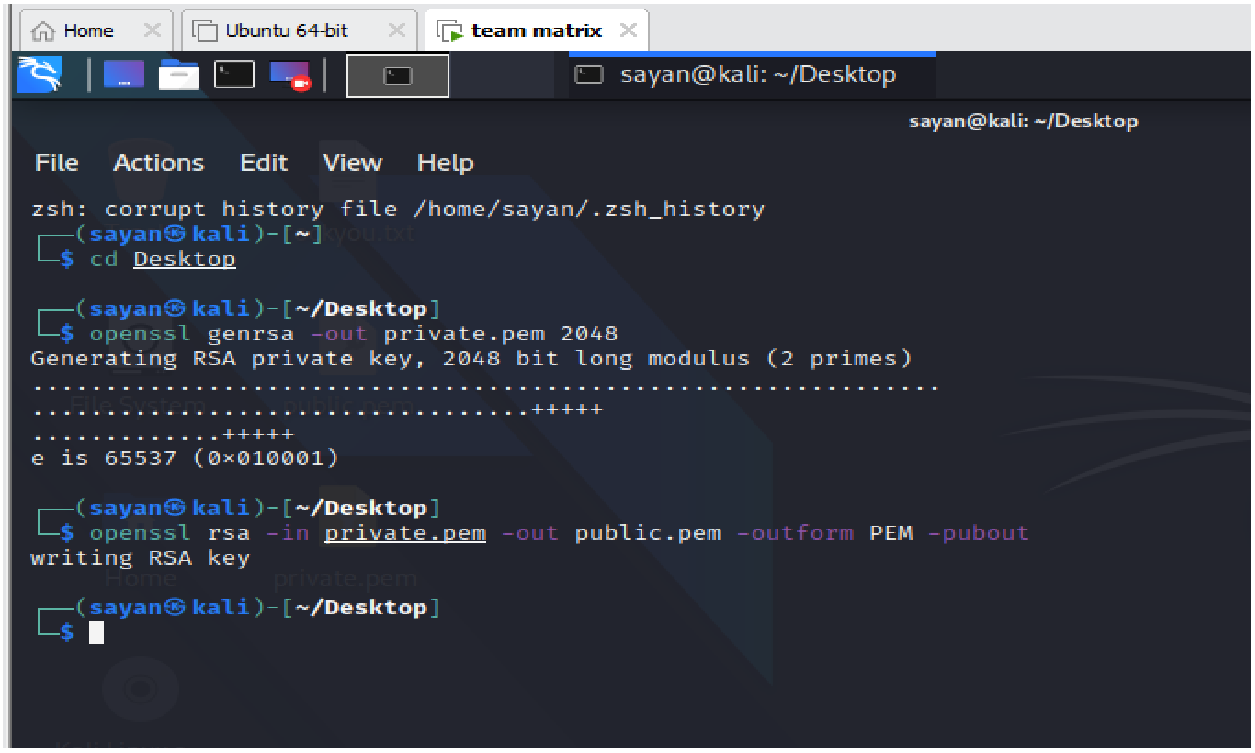
Methodology:

Step1: At first open the kali terminal:



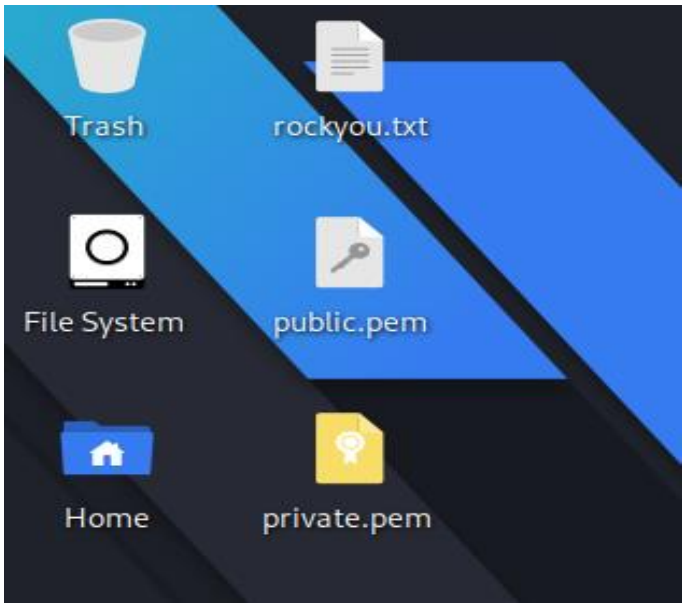
```
zsh: corrupt history file /home/sayan/.zsh_history
┌──(sayan㉿kali)-[~]
└─$ 
```

Step2: I have access to confidential information. In desktop, a pem file is created.

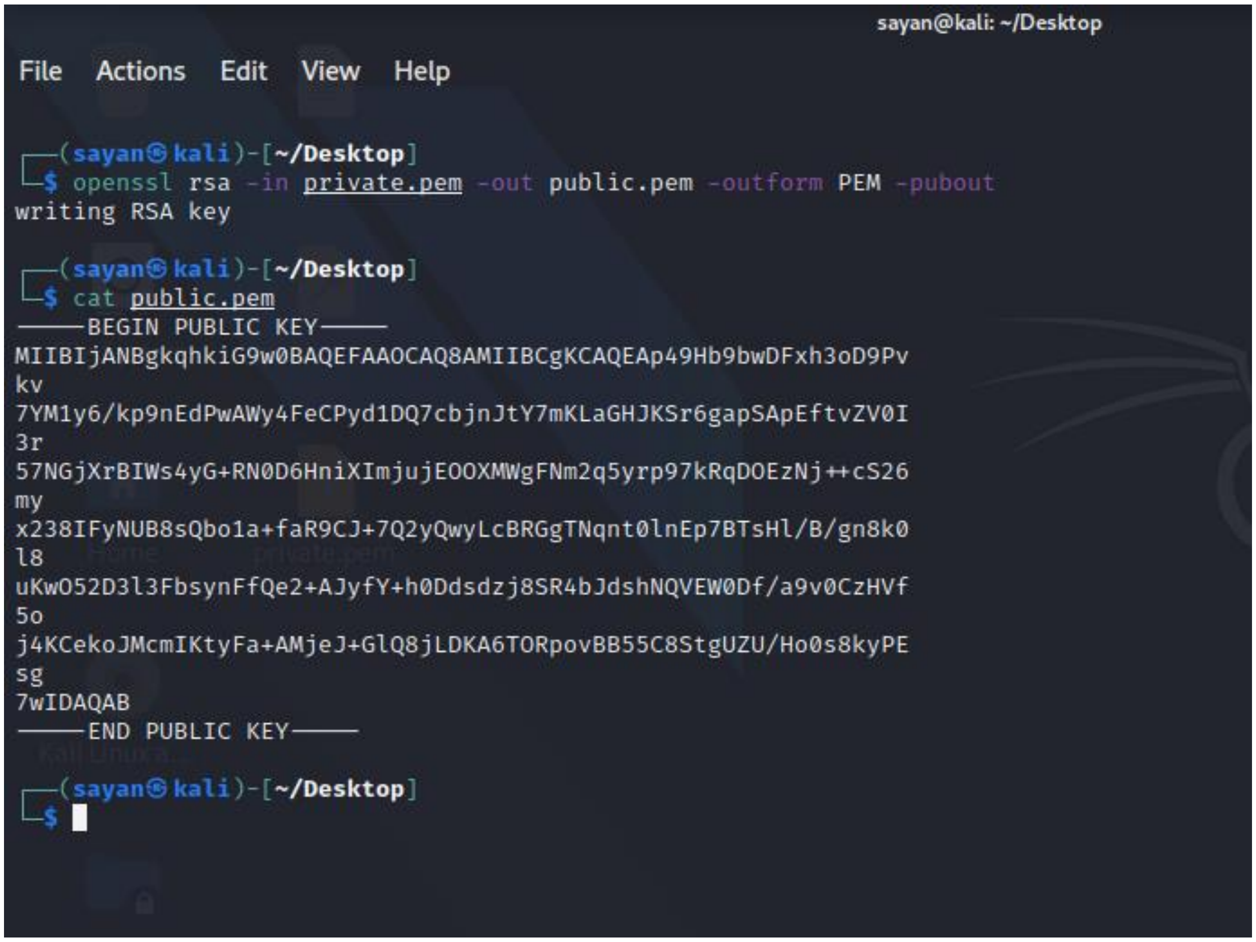


```
zsh: corrupt history file /home/sayan/.zsh_history
┌──(sayan㉿kali)-[~]
└─$ cd Desktop

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl genrsa -out private.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
..............................................................
.................................+++++
.............+++++
e is 65537 (0×010001)

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl rsa -in private.pem -out public.pem -outform PEM -pubout
writing RSA key

┌──(sayan㉿kali)-[~/Desktop]
└─$ 
```

Step3: Let’s open the private key

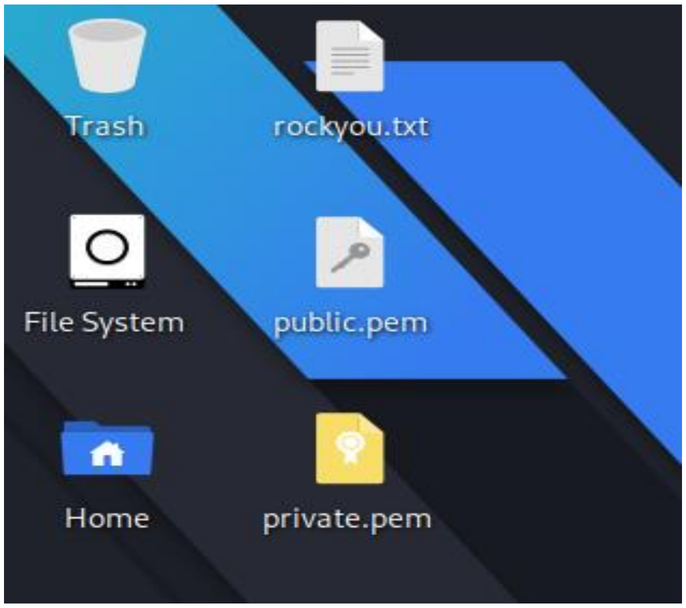


Step4: Create a public key now.

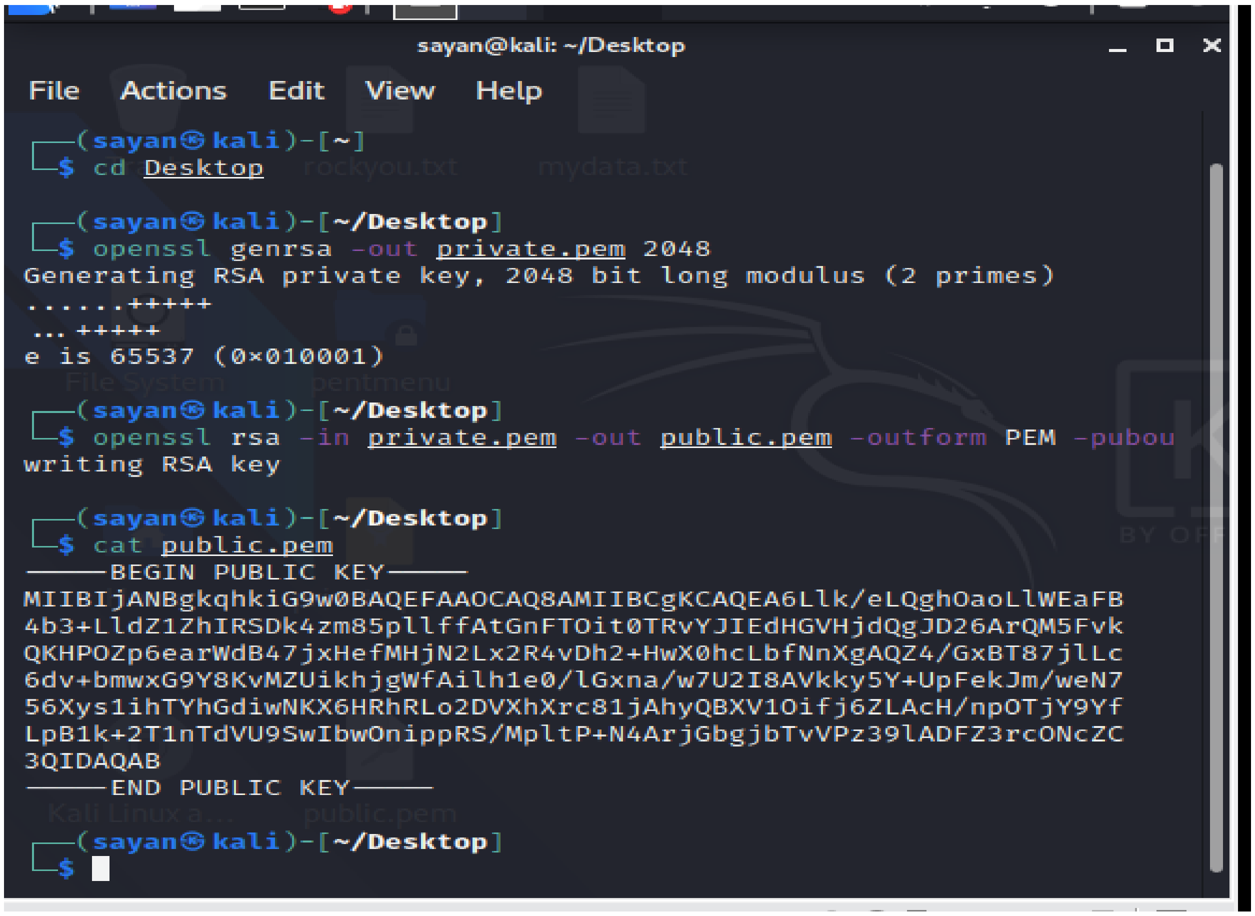


```
┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl rsa -in private.pem -out public.pem -outform PEM -pubout
writing RSA key

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAp49Hb9bwDFxh3oD9Pv
kv
7YM1y6/kp9nEdPwAWy4FeCPyd1DQ7cbjnJtY7mKLaGHJKSr6gapSApEftvZV0I
3r
57NGjXrBIWs4yG+RN0D6HniXImjujEOOXMWgFNm2q5yrp97kRqDOEzNj++cS26
my
x238IFyNUB8sQbo1a+faR9CJ+7Q2yQwyLcBRGgTNqnt0lnEp7BTsHl/B/gn8k0
l8
uKwO52D3l3FbsynFfQe2+AJyfY+h0Ddsdzj8SR4bJdshNQVEW0Df/a9v0CzHVf
5o
j4KCekoJMcmIKtyFa+AMjeJ+GlQ8jLDKA6TORpovBB55C8StgUZU/Ho0s8kyPE
sg
7wIDAQAB
-----END PUBLIC KEY-----

┌──(sayan㉿kali)-[~/Desktop]
└─$ 
```

Step5: Another file, public.pem, is created in the desktop.

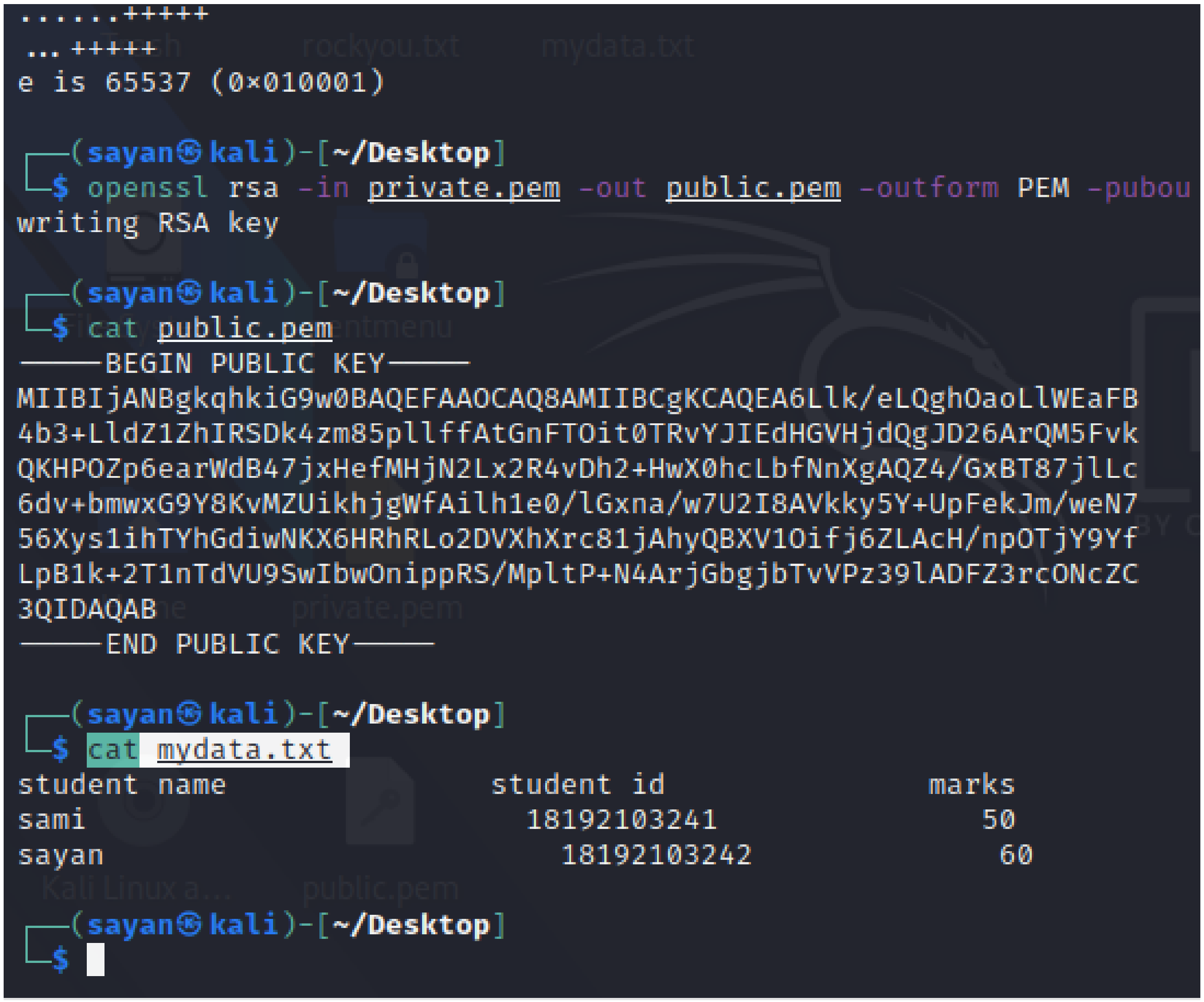


Step6: All right, let's crack it open.

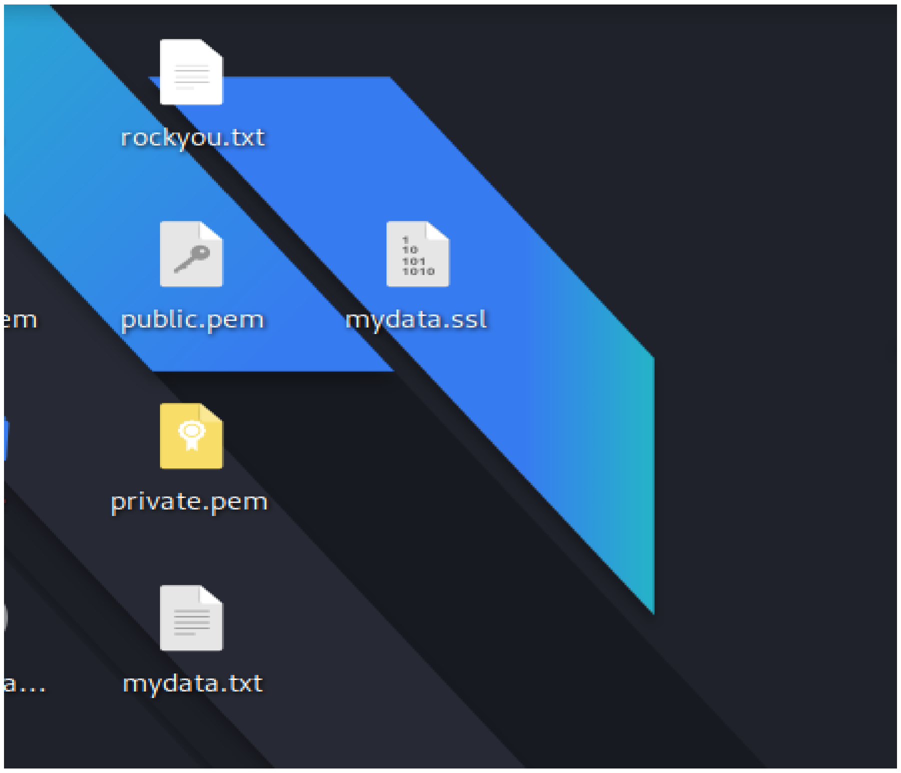


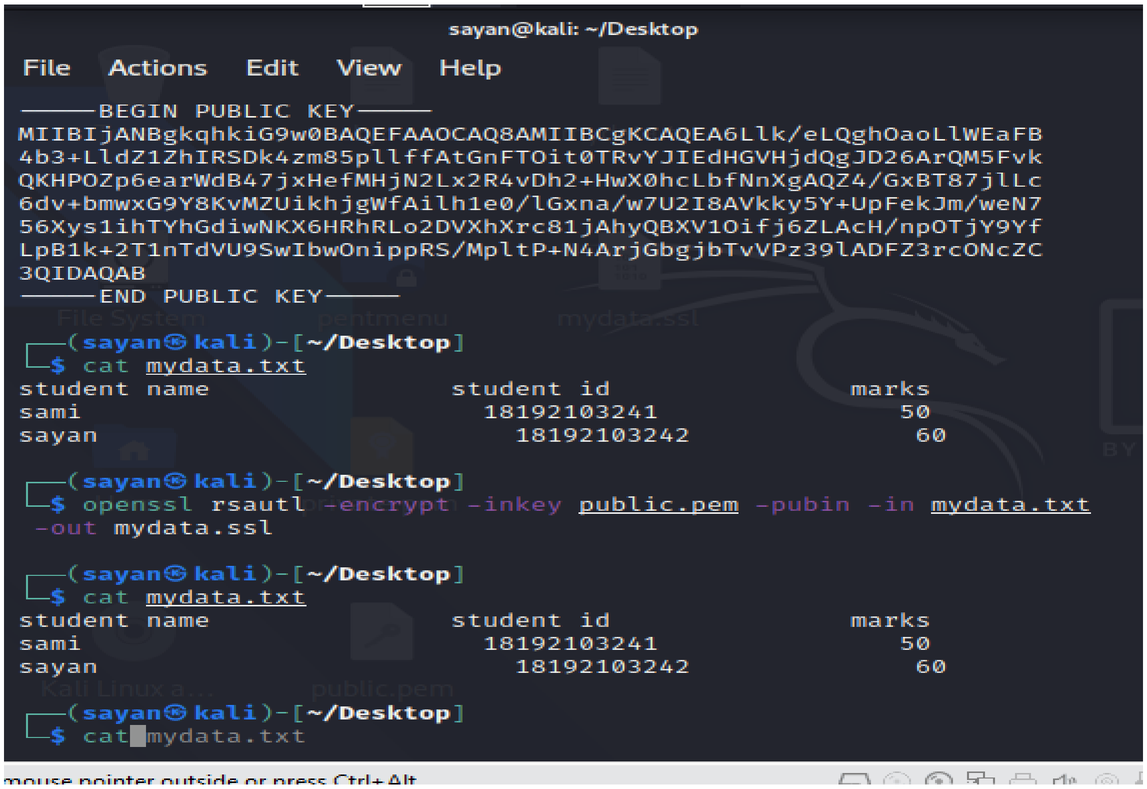
```
┌──(sayan㉿kali)-[~]
└─$ cd Desktop

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl genrsa -out private.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
......+++++
... +++++
e is 65537 (0×010001)

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl rsa -in private.pem -out public.pem -outform PEM -pubou
writing RSA key

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6Llk/eLQghOaoLlWEaFB
4b3+LldZ1ZhIRSDk4zm85pllffAtGnFTOit0TRvYJIEdHGVHjdQgJD26ArQM5Fvk
QKHPOZp6earWdB47jxHefMHjN2Lx2R4vDh2+HwX0hcLbfNnXgAQZ4/GxBT87jlLc
6dv+bmwxG9Y8KvMZUikhjgWfAilh1e0/lGxna/w7U2I8AVkky5Y+UpFekJm/weN7
56Xys1ihTYhGdiwNKX6HRhRLo2DVXhXrc81jAhyQBXV1Oifj6ZLAcH/npOTjY9Yf
LpB1k+2T1nTdVU9SwIbwOnippRS/MpltP+N4ArjGbgjbTvVPz39lADFZ3rcONcZC
3QIDAQAB
-----END PUBLIC KEY-----

┌──(sayan㉿kali)-[~/Desktop]
└─$ 
```

Step 7: I'll check out, rather at random, a large number of public keys. At this point, we have already produced two keys. The next step is to learn how to encrypt and decode files. The method described above is what's known as asymmetric encryption. First, let's create a text file and call it mydata.txt so that we may examine the contents of the file.



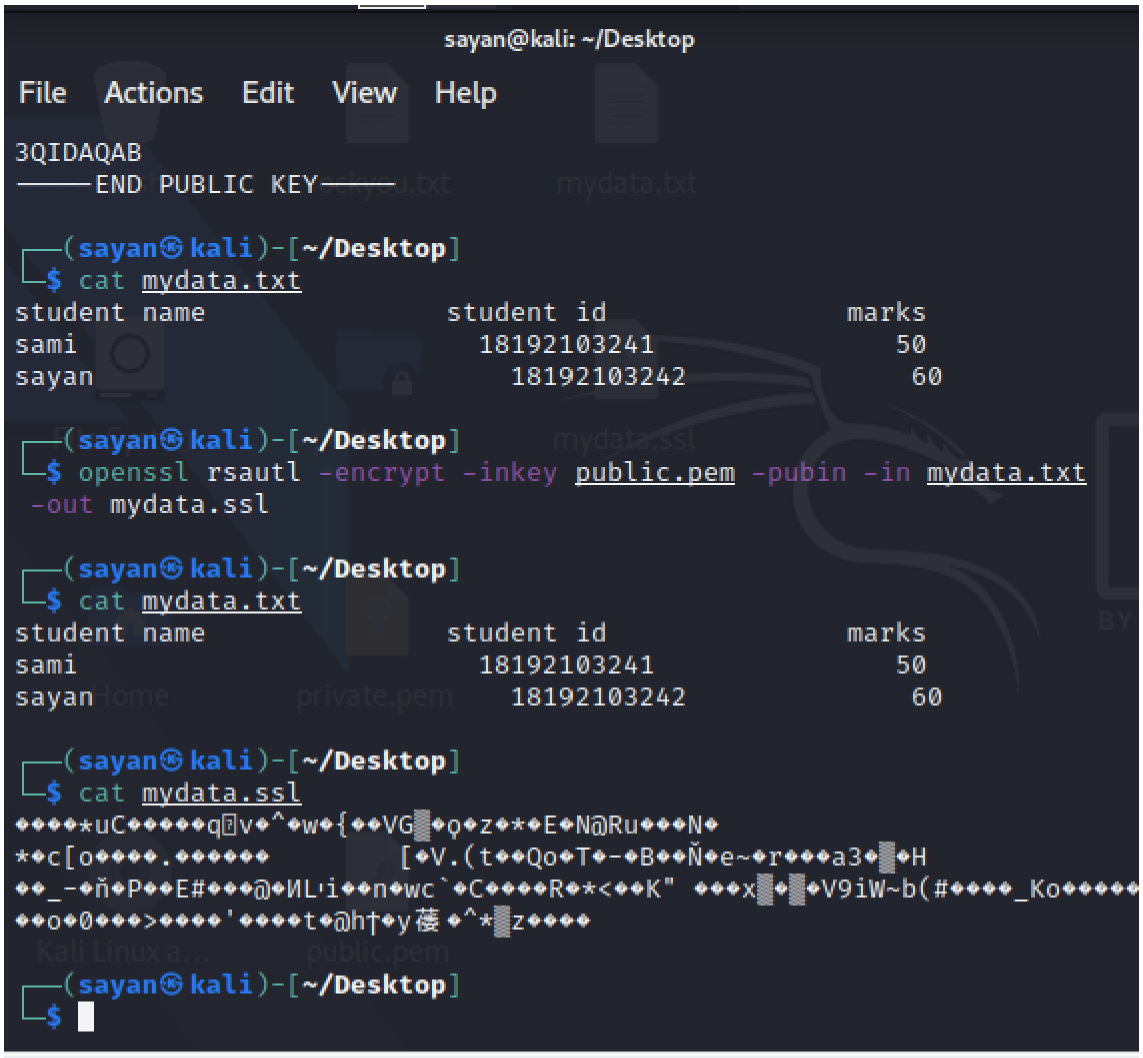
```
......+++++
...+++++
e is 65537 (0×010001)

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl rsa -in private.pem -out public.pem -outform PEM -pubou
writing RSA key

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6Llk/eLQghOaoLlWEaFB
4b3+LldZ1ZhIRSDk4zm85pllffAtGnFTOit0TRvYJIEdHGVHjdQgJD26ArQM5Fvk
QKHPOZp6earWdB47jxHefMHjN2Lx2R4vDh2+HwX0hcLbfNnXgAQZ4/GxBT87jlLc
6dv+bmwxG9Y8KvMZUikhjgWfAilh1e0/lGxna/w7U2I8AVkky5Y+UpFekJm/weN7
56Xys1ihTYhGdiwNKX6HRhRLo2DVXhXrc81jAhyQBXV1Oifj6ZLAcH/npOTjY9Yf
LpB1k+2T1nTdVU9SwIbwOnippRS/MpltP+N4ArjGbgjbTvVPz39lADFZ3rcONcZC
3QIDAQAB
-----END PUBLIC KEY-----

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.txt
student name                student id              marks
sami                          18192103241              50
sayan                           18192103242             60

┌──(sayan㉿kali)-[~/Desktop]
└─$ 
```





```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6Llk/eLQghOaoLlWEaFB
4b3+LldZ1ZhIRSDk4zm85pllffAtGnFTOit0TRvYJIEdHGVHjdQgJD26ArQM5Fvk
QKHPOZp6earWdB47jxHefMHjN2Lx2R4vDh2+HwX0hcLbfNnXgAQZ4/GxBT87jlLc
6dv+bmwxG9Y8KvMZUikhjgWfAilh1e0/lGxna/w7U2I8AVkky5Y+UpFekJm/weN7
56Xys1ihTYhGdiwNKX6HRhRLo2DVXhXrc81jAhyQBXV1Oifj6ZLAcH/npOTjY9Yf
LpB1k+2T1nTdVU9SwIbwOnippRS/MpltP+N4ArjGbgjbTvVPz39lADFZ3rcONcZC
3QIDAQAB
-----END PUBLIC KEY-----

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.txt
student name               student id               marks
sami                         18192103241              50
sayan                          18192103242              60

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl rsautl -encrypt -inkey public.pem -pubin -in mydata.txt
 -out mydata.ssl

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.txt
student name               student id               marks
sami                         18192103241              50
sayan                          18192103242              60

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.txt
```

mouse pointer outside or press Ctrl+Alt



sayan@kali: ~/Desktop

File Actions Edit View Help

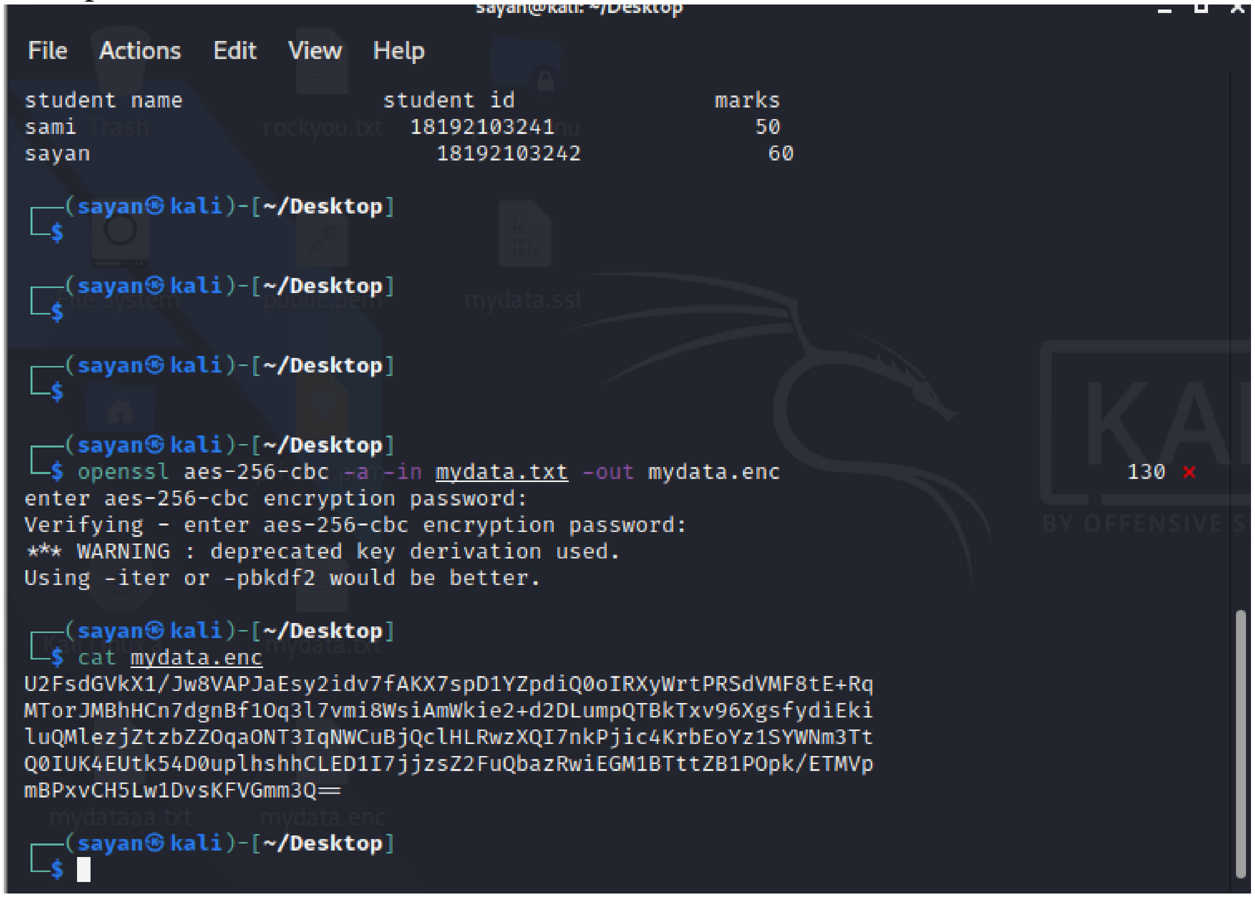
```
3QIDAQAB
-----END PUBLIC KEY-----

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.txt
student name                  student id               marks
sami                            18192103241               50
sayan                             18192103242               60

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl rsautl -encrypt -inkey public.pem -pubin -in mydata.txt
 -out mydata.ssl

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.txt
student name                  student id               marks
sami                            18192103241               50
sayan                             18192103242               60

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.ssl
◆◆◆◆*uC◆◆◆◆◆◆q⍰v◆^◆w◆{◆◆VG▒◆ọ◆z◆*◆E◆N@Ru◆◆◆N◆
*◆c[o◆◆◆◆◆.◆◆◆◆◆◆        [◆V.(t◆◆Qo◆T◆-◆B◆◆Ň◆e~◆r◆◆◆◆a3◆▒◆H
◆◆_-◆ň◆P◆◆E#◆◆◆@◆ИLˈi◆◆n◆wc`◆C◆◆◆◆R◆*<◆◆K" ◆◆◆x▒◆▒◆V9iW~b(#◆◆◆◆_Ko◆◆◆◆◆◆
◆◆o◆0◆◆◆>◆◆◆◆'◆◆◆◆t◆@h†◆y薐◆^*▒z◆◆◆◆

┌──(sayan㉿kali)-[~/Desktop]
└─$ 
```

Step 8:

The Advanced Encryption Standard (AES) is the first and only cipher that is available to the general public that has been given the green light for use in protecting top secret material by the United States National Security Agency (NSA). Rijndael was the original name given to AES, and it was named after the Belgian cryptographers Vincent Rijmen and Joan Daemen who were responsible for its development.

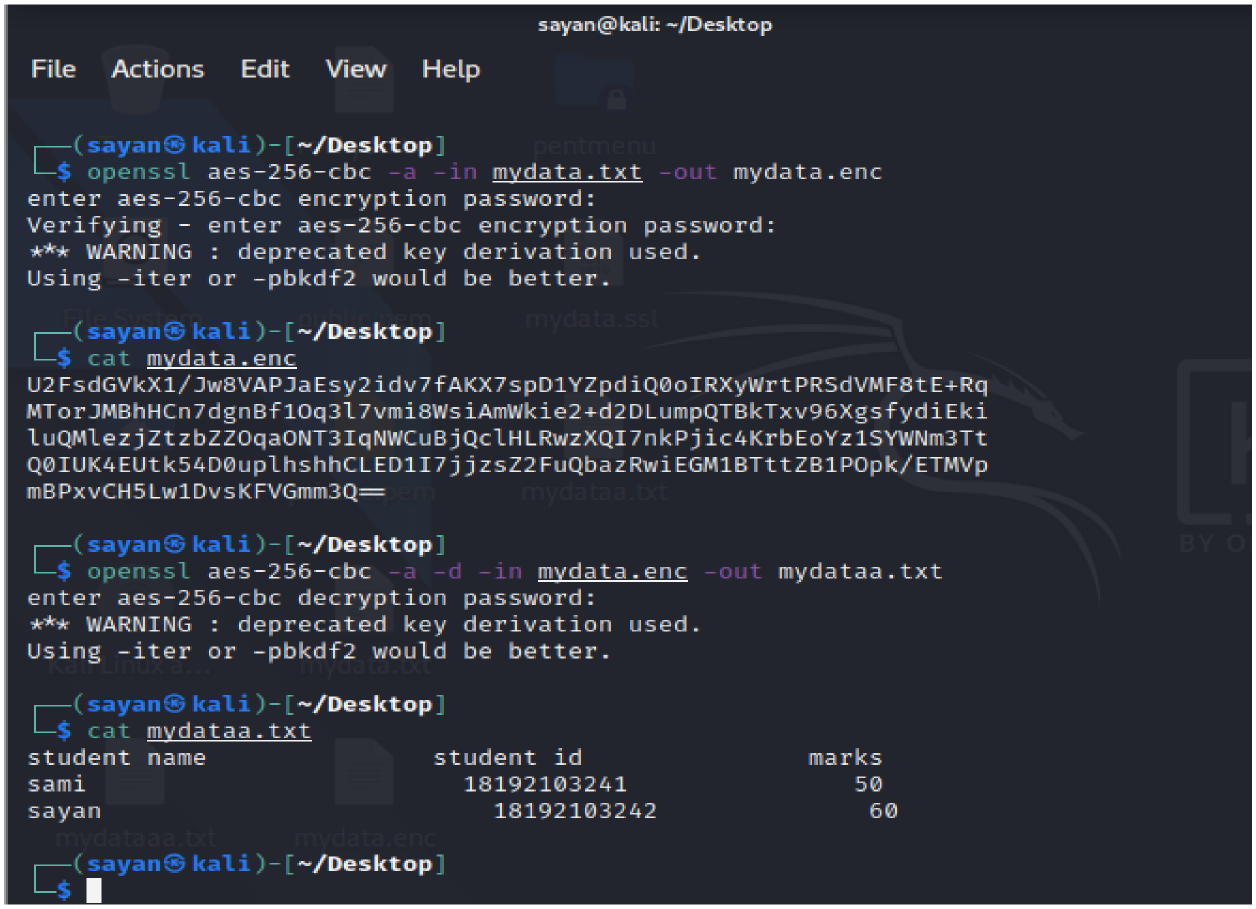
The explanation for how symmetric key encryption works can be found in the following illustration:

Step 9: Let's encrypt a file using the ASE-256 key, such is mydata.txt for example.



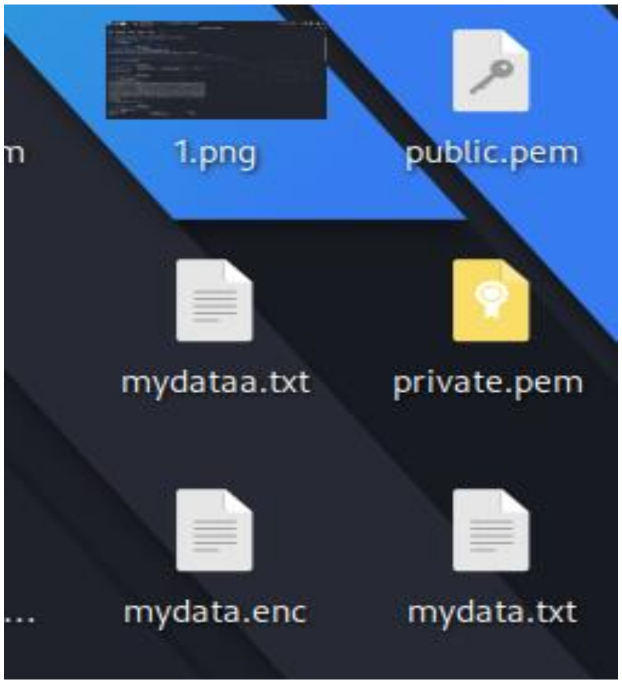
```
student name                student id              marks
sami                          18192103241              50
sayan                           18192103242              60

┌──(sayan㉿kali)-[~/Desktop]
└─$ 

┌──(sayan㉿kali)-[~/Desktop]
└─$ 

┌──(sayan㉿kali)-[~/Desktop]
└─$ 

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl aes-256-cbc -a -in mydata.txt -out mydata.enc                          130 ✕
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.enc
U2FsdGVkX1/Jw8VAPJaEsy2idv7fAKX7spD1YZpdiQ0oIRXyWrtPRSdVMF8tE+Rq
MTorJMBhHCn7dgnBf1Oq3l7vmi8WsiAmWkie2+d2DLumpQTBkTxv96XgsfydiEki
luQMlezjZtzbZZOqaONT3IqNWCuBjQclHLRwzXQI7nkPjic4KrbEoYz1SYWNm3Tt
Q0IUK4EUtk54D0uplhshhCLED1I7jjzsZ2FuQbazRwiEGM1BTttZB1POpk/ETMVp
mBPxvCH5Lw1DvsKFVGmm3Q==

┌──(sayan㉿kali)-[~/Desktop]
└─$ 
```

Step10:



```
┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl aes-256-cbc -a -in mydata.txt -out mydata.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydata.enc
U2FsdGVkX1/Jw8VAPJaEsy2idv7fAKX7spD1YZpdiQ0oIRXyWrtPRSdVMF8tE+Rq
MTorJMBhHCn7dgnBf1Oq3l7vmi8WsiAmWkie2+d2DLumpQTBkTxv96XgsfydiEki
luQMlezjZtzbZZOqaONT3IqNWCuBjQclHLRwzXQI7nkPjic4KrbEoYz1SYWNm3Tt
Q0IUK4EUtk54D0uplhshhCLED1I7jjzsZ2FuQbazRwiEGM1BTttZB1POpk/ETMVp
mBPxvCH5Lw1DvsKFVGmm3Q==

┌──(sayan㉿kali)-[~/Desktop]
└─$ openssl aes-256-cbc -a -d -in mydata.enc -out mydataa.txt
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

┌──(sayan㉿kali)-[~/Desktop]
└─$ cat mydataa.txt
student name                student id               marks
sami                          18192103241              50
sayan                           18192103242              60

┌──(sayan㉿kali)-[~/Desktop]
└─$ 
```

.


1.png
public.pem
mydataa.txt
private.pem
mydata.enc
mydata.txt