7481 Take Home Exam

Design & Documentation

Eric Wu, A00961904
Hong Kit, Wu A00968591

BTech 2020 Winter

Overview 2

Propose system 2

Example scenario 4

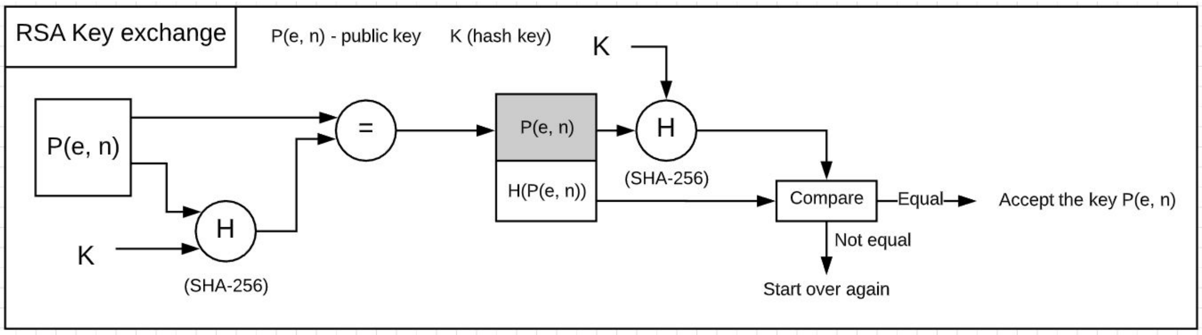
Captures and Run-through 5

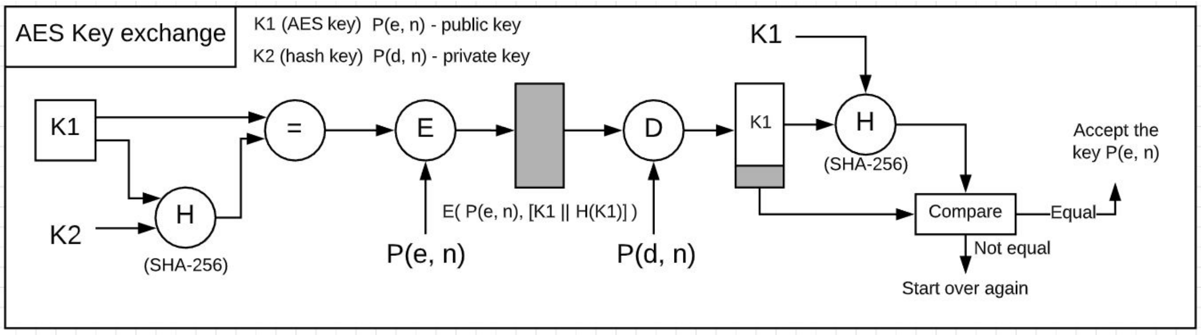
Overview

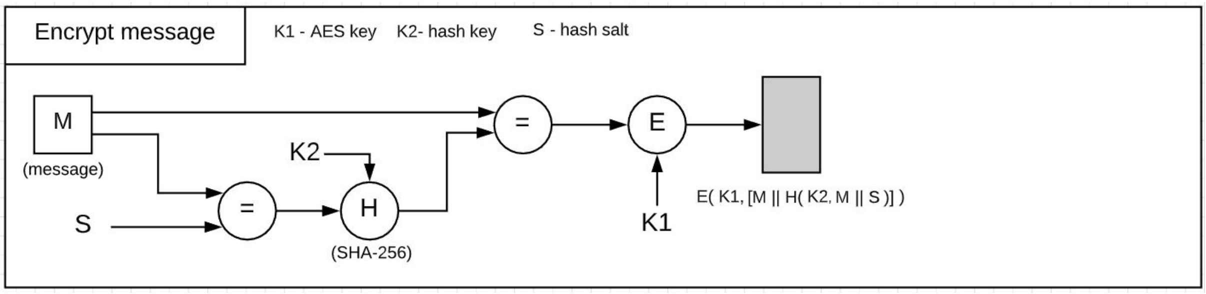
Propose an experimental system, using the algorithms and techniques we have learned in this course. In this system, there are two parties, Bob and Alice, who live far away but wish to communicate with each other. Below are some possible elements of the system (* bold text are elements used)

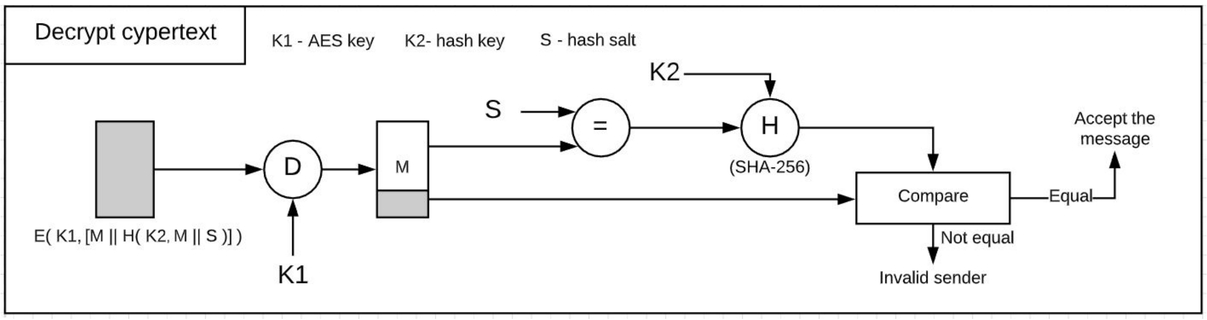
a) **Symmetric key**
b) **Asymmetric keys**
c) **Passphrase (for private key)**
d) **Keyed hash**
e) **Salted hash**
f) Some enemy (Eve)
g) Other

Propose system









In this system, we are assuming that the private key generation involves adding passphrase. The sample system captures all three aspects of cryptography - **confidentiality**, **integrity** and **authentication**. The hash functions are used to ensure that the communication happens only between the intended sender and receiver; It also ensures that the received data has not been altered when transmitted over the channel. The “Salt” added to the hashing procedure strengthens the “authentication” property even more.

Both symmetric and asymmetric are used in this system. Before encrypting the plaintext, the plaintext is first hashed with SHA-256 algorithm along with “Salt”. The symmetric key (AES) is used to encrypt the hashed output to produce ciphertext. The asymmetric key is only used when exchanging the AES key.

Example scenario

Assumption

- Alice is the one who initiates the communication
- Alice and Bob both agree in advance on the hash algorithm (SHA-256), hash salt, and hash key.

Key exchange

1. Bob use the passphrase to generate an RSA **P(e, n)** (public key) and **P(d, n)** (private key)
2. Alice use the passphrase to generates an RSA **P(e, n)** and **P(d, n)**
3. Bob hash his public key with hash key "**H(K, P(e, n))**" and mail it to Alice
4. Alice hash her public key with hash key "**H(K, P(e, n))**" and mail it to Bob
5. Alice generates a AES key "**K1**", hash it with hash key "**K2**", encrypt it with Bob's public key "**P(e, n)**", and mail it to Bob "**E(P(e, n), [K1 || H(K2, K1)])**"
6. Bob receives encrypted AES key, decrypt it with his private key "**P(d, n)**", check if hash valid, and retrieve the decrypted AES key "**K1**"

Begin transmission (one side)

1. Alice append "**salt**" to her secret message "**S || M**"
2. Alice hash the output with hash key "**K2**" and combine it with original message to produce "**M || H(K2, S || M)**"
3. Alice then encrypt the output with both Bob's public key and passphrase to produce "**E(P(e, n), [M || H(K2, S || M)])**", and mail it to Bob
4. Bob received the ciphertext, decrypt with the AES key "**K1**"
5. Bob then run the message through the hash algorithm along with "**salt**" and hash key "**K2**" to check if the message is indeed from Alice
6. If the hash check pass, then Bob has received the actual secret message from Alice
7. Transmission completed

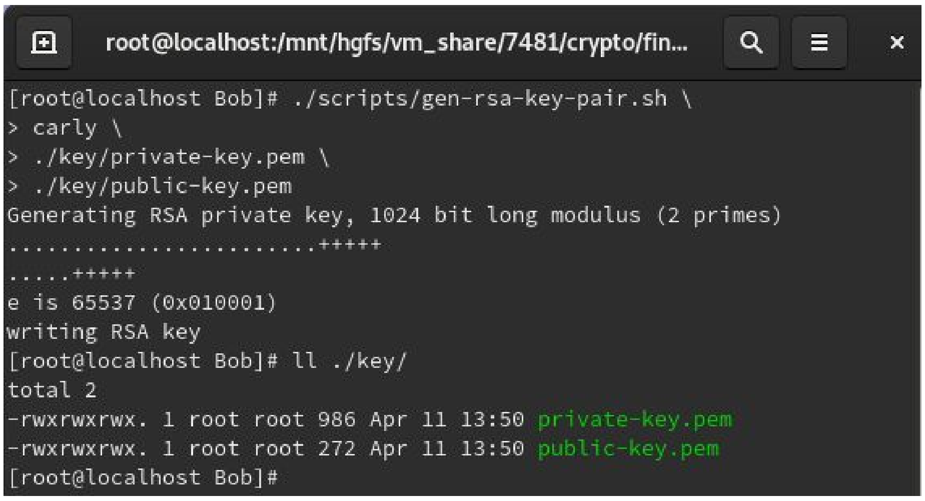
Captures and Run-through

Alice and Bob both agree in advance on the

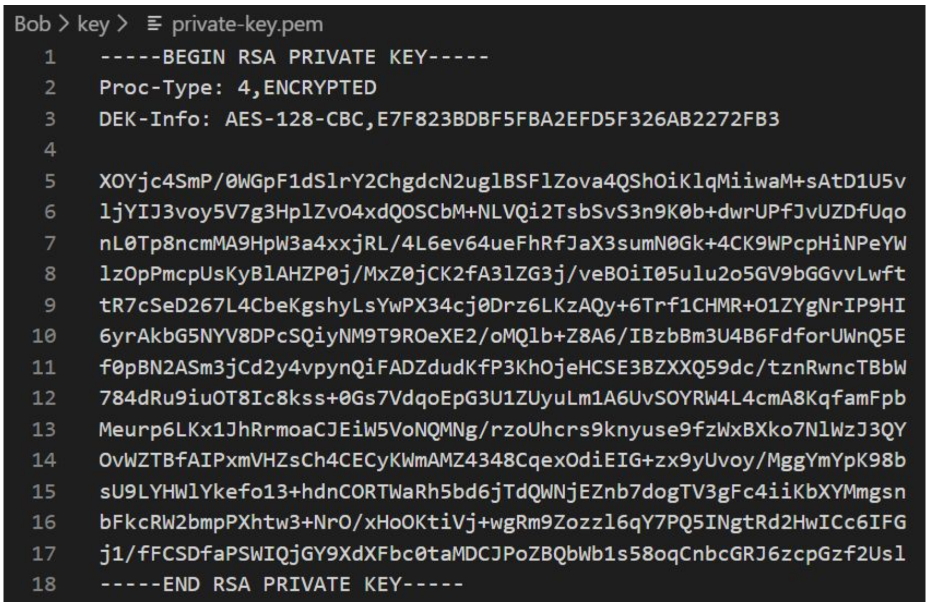
- hash algorithm (SHA-256)
- hash salt and the hash key (refer to README.md).

We are using linux command “**cp**” to simulate mailing behaviour

1) Bob use the passphrase “**carly**” to generate an RSA **P(e, n)** (public key) and **P(d, n)** (private key)

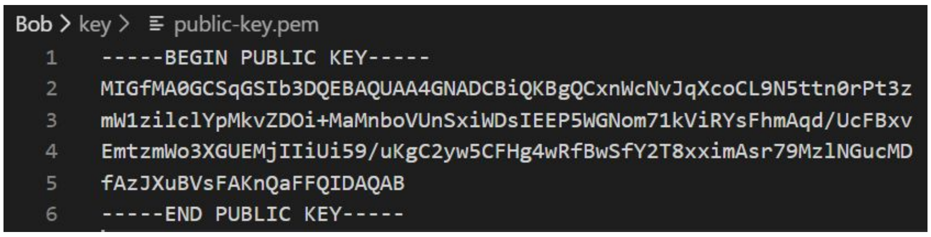


```
[root@localhost Bob]# ./scripts/gen-rsa-key-pair.sh \
> carly \
> ./key/private-key.pem \
> ./key/public-key.pem
Generating RSA private key, 1024 bit long modulus (2 primes)
........................+++++
.....+++++
e is 65537 (0x010001)
writing RSA key
[root@localhost Bob]# ll ./key/
total 2
-rwxrwxrwx. 1 root root 986 Apr 11 13:50 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:50 public-key.pem
[root@localhost Bob]#
```



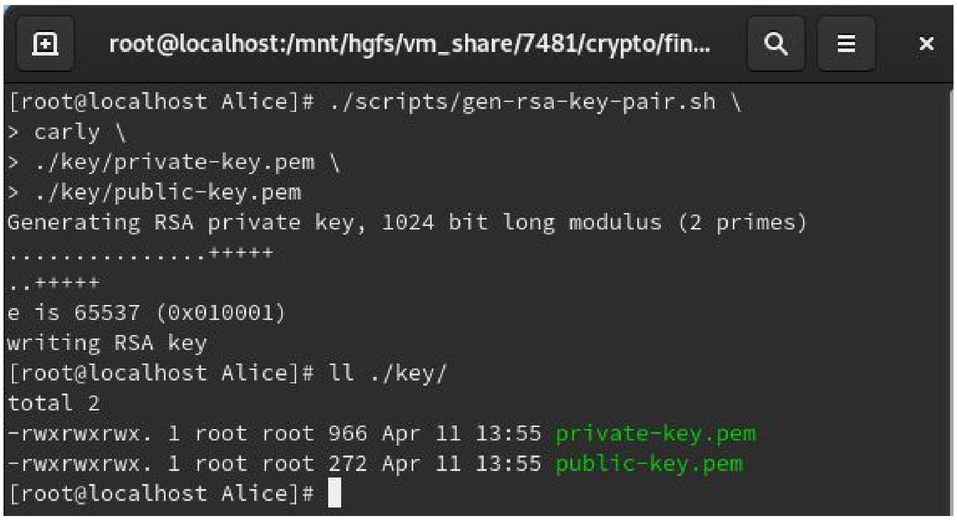
```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,E7F823BDBF5FBA2EFD5F326AB2272FB3

XOYjc4SmP/0WGpF1dSlrY2ChgdcN2uglBSFlZova4QShOiKlqMiiwaM+sAtD1U5v
ljYIJ3voy5V7g3HplZvO4xdQOSCbM+NLVQi2TsbSvS3n9K0b+dwrUPfJvUZDfUqo
nL0Tp8ncmMA9HpW3a4xxjRL/4L6ev64ueFhRfJaX3sumN0Gk+4CK9WPcpHiNPeYW
lzOpPmcpUsKyBlAHZP0j/MxZ0jCK2fA3lZG3j/veBOiI05ulu2o5GV9bGGvvLwft
tR7cSeD267L4CbeKgshyLsYwPX34cj0Drz6LKzAQy+6Trf1CHMR+O1ZYgNrIP9HI
6yrAkbG5NYV8DPcSQiyNM9T9ROeXE2/oMQlb+Z8A6/IBzbBm3U4B6FdforUWnQ5E
f0pBN2ASm3jCd2y4vpynQiFADZdudKfP3KhOjeHCSE3BZXXQ59dc/tznRwncTBbW
784dRu9iuOT8Ic8kss+0Gs7VdqoEpG3U1ZUyuLm1A6UvSOYRW4L4cmA8KqfamFpb
Meurp6LKx1JhRrmoaCJEiW5VoNQMNg/rzoUhcrs9knyuse9fzWxBXko7NlWzJ3QY
OvWZTBfAIPxmVHZsCh4CECyKWmAMZ4348CqexOdiEIG+zx9yUvoy/MggYmYpK98b
sU9LYHWlYkefo13+hdnCORTWaRh5bd6jTdQWNjEZnb7dogTV3gFc4iiKbXYMmgsn
bFkcRW2bmpPXhtw3+NrO/xHoOKtiVj+wgRm9Zozzl6qY7PQ5INgtRd2HwICc6IFG
j1/fFCSDfaPSWIQjGY9XdXFbc0taMDCJPoZBQbWb1s58oqCnbcGRJ6zcpGzf2Usl
-----END RSA PRIVATE KEY-----
```

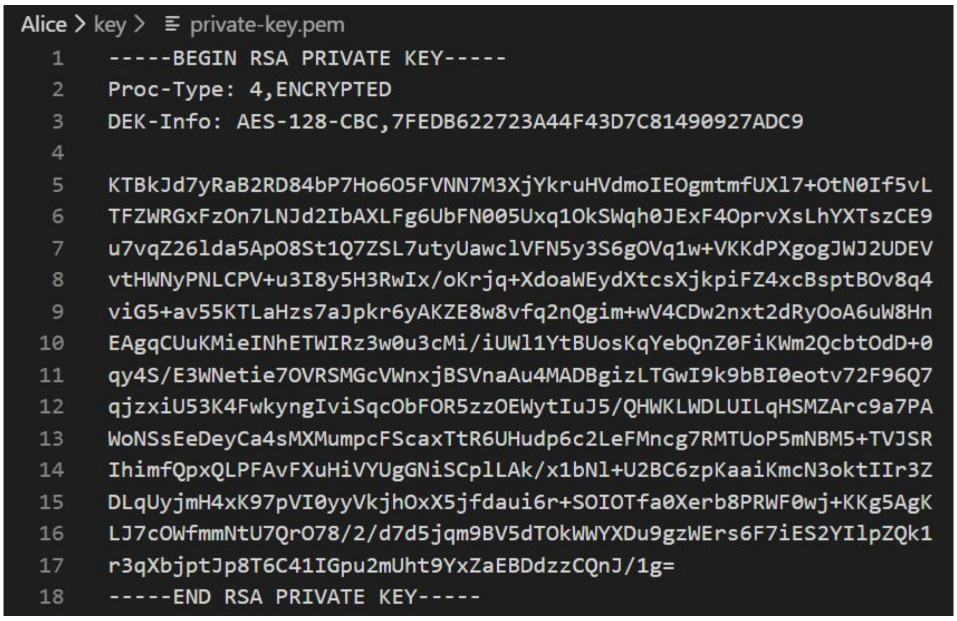


```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCxnWcNvJqXcoCL9N5ttn0rPt3z
mW1zilclYpMkvZDOi+MaMnboVUnSxiWDsIEEP5WGNom71kViRYsFhmAqd/UcFBxv
EmtzmWo3XGUEMjIIiUi59/uKgC2yw5CFHg4wRfBwSfY2T8xximAsr79MzlNGucMD
fAzJXuBVsFAKnQaFFQIDAQAB
-----END PUBLIC KEY-----
```

2) Alice use the passphrase “**carly**” to generate an RSA **P(e, n)** (public key) and **P(d, n)** (private key)

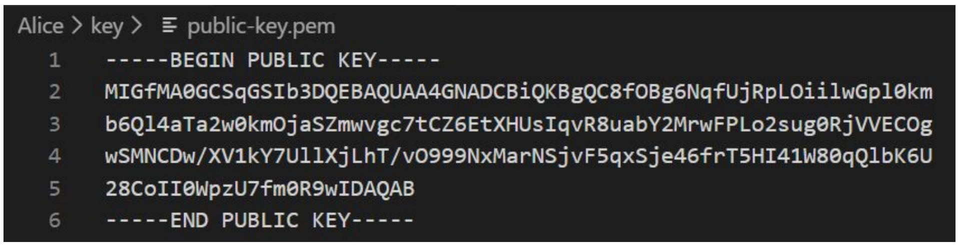


```
[root@localhost Alice]# ./scripts/gen-rsa-key-pair.sh \
> carly \
> ./key/private-key.pem \
> ./key/public-key.pem
Generating RSA private key, 1024 bit long modulus (2 primes)
..............+++++
..+++++
e is 65537 (0x010001)
writing RSA key
[root@localhost Alice]# ll ./key/
total 2
-rwxrwxrwx. 1 root root 966 Apr 11 13:55 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:55 public-key.pem
[root@localhost Alice]#
```



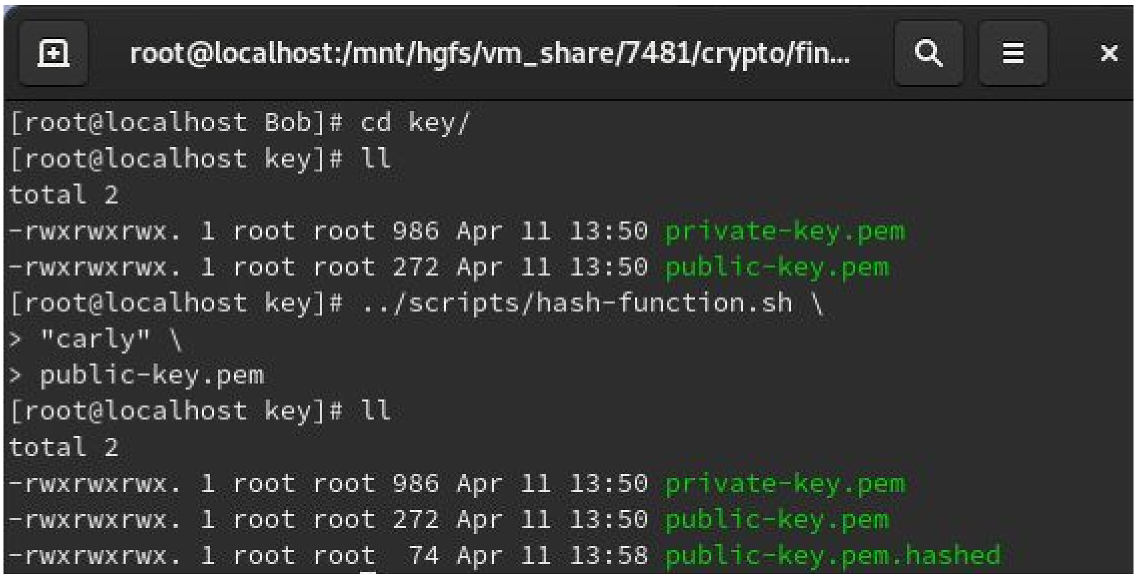
```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,7FEDB622723A44F43D7C81490927ADC9

KTBkJd7yRaB2RD84bP7Ho6O5FVNN7M3XjYkruHVdmoIEOgmtmfUXl7+OtN0If5vL
TFZWRGxFzOn7LNJd2IbAXLFg6UbFN005Uxq1OkSWqh0JExF4OprvXsLhYXTszCE9
u7vqZ26lda5ApO8St1Q7ZSL7utyUawclVFN5y3S6gOVq1w+VKKdPXgogJWJ2UDEV
vtHWNyPNLCPV+u3I8y5H3RwIx/oKrjq+XdoaWEydXtcsXjkpiFZ4xcBsptBOv8q4
viG5+av55KTLaHzs7aJpkr6yAKZE8w8vfq2nQgim+wV4CDw2nxt2dRyOoA6uW8Hn
EAgqCUuKMieINhETWIRz3w0u3cMi/iUWl1YtBUosKqYebQnZ0FiKWm2QcbtOdD+0
qy4S/E3WNetie7OVRSMGcVWnxjBSVnaAu4MADBgizLTGwI9k9bBI0eotv72F96Q7
qjzxiU53K4FwkyngIviSqcObFOR5zzOEWytIuJ5/QHWKLWDLUILqHSMZArc9a7PA
WoNSsEeDeyCa4sMXMumpcFScaxTtR6UHudp6c2LeFMncg7RMTUoP5mNBM5+TVJSR
IhimfQpxQLPFAvFXuHiVYUgGNiSCplLAk/x1bNl+U2BC6zpKaaiKmcN3oktIIr3Z
DLqUyjmH4xK97pVI0yyVkjhOxX5jfdaui6r+SOIOTfa0Xerb8PRWF0wj+KKg5AgK
LJ7cOWfmmNtU7QrO78/2/d7d5jqm9BV5dTOkWWYXDu9gzWErs6F7iES2YIlpZQk1
r3qXbjptJp8T6C41IGpu2mUht9YxZaEBDdzzCQnJ/1g=
-----END RSA PRIVATE KEY-----
```

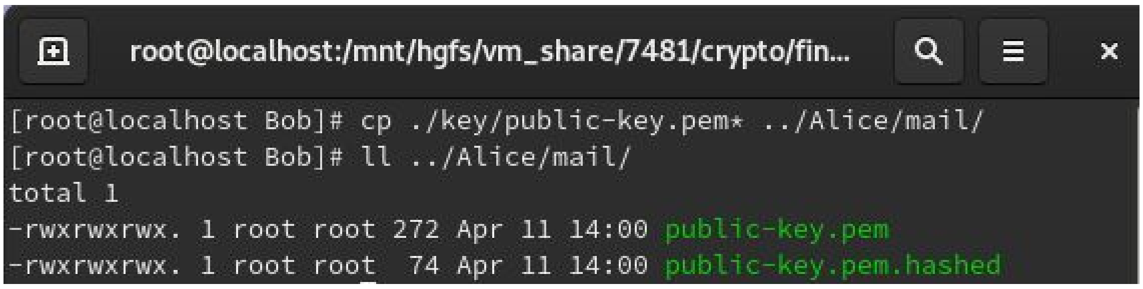


```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC8fOBg6NqfUjRpLOiilwGpl0km
b6Ql4aTa2w0kmOjaSZmwvgc7tCZ6EtXHUsIqvR8uabY2MrwFPLo2sug0RjVVECOg
wSMNCDw/XV1kY7UllXjLhT/vO999NxMarNSjvF5qxSje46frT5HI41W80qQlbK6U
28CoII0WpzU7fm0R9wIDAQAB
-----END PUBLIC KEY-----
```

3) Bob hash his public key with hash key "**carly**" and mail both the original and hash public key to Alice

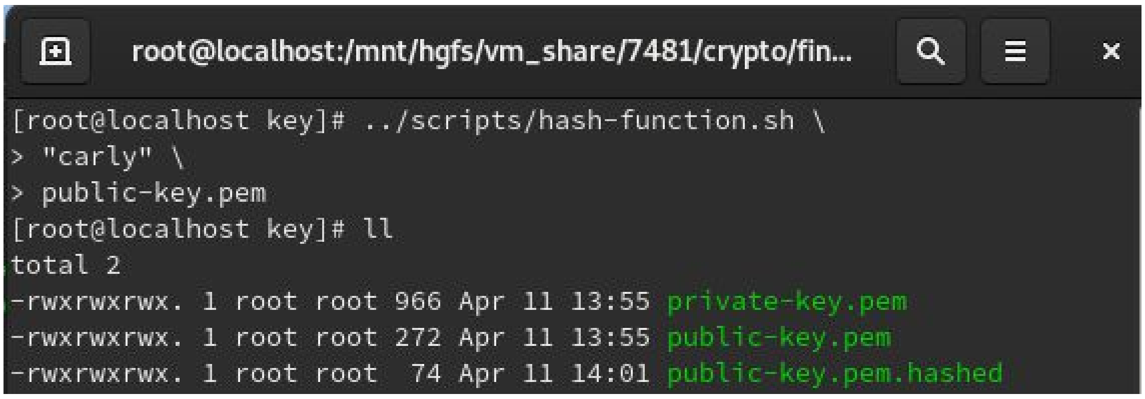


```
[root@localhost Bob]# cd key/
[root@localhost key]# ll
total 2
-rwxrwxrwx. 1 root root 986 Apr 11 13:50 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:50 public-key.pem
[root@localhost key]# ../scripts/hash-function.sh \
> "carly" \
> public-key.pem
[root@localhost key]# ll
total 2
-rwxrwxrwx. 1 root root 986 Apr 11 13:50 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:50 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 13:58 public-key.pem.hashed
```

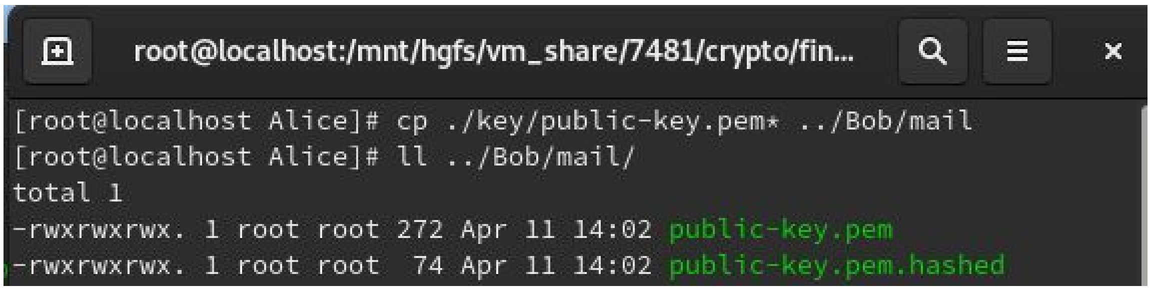


```
[root@localhost Bob]# cp ./key/public-key.pem* ../Alice/mail/
[root@localhost Bob]# ll ../Alice/mail/
total 1
-rwxrwxrwx. 1 root root 272 Apr 11 14:00 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:00 public-key.pem.hashed
```

4) Alice hash her public key with hash key "**carly**" and mail both the original and hash public key to Bob

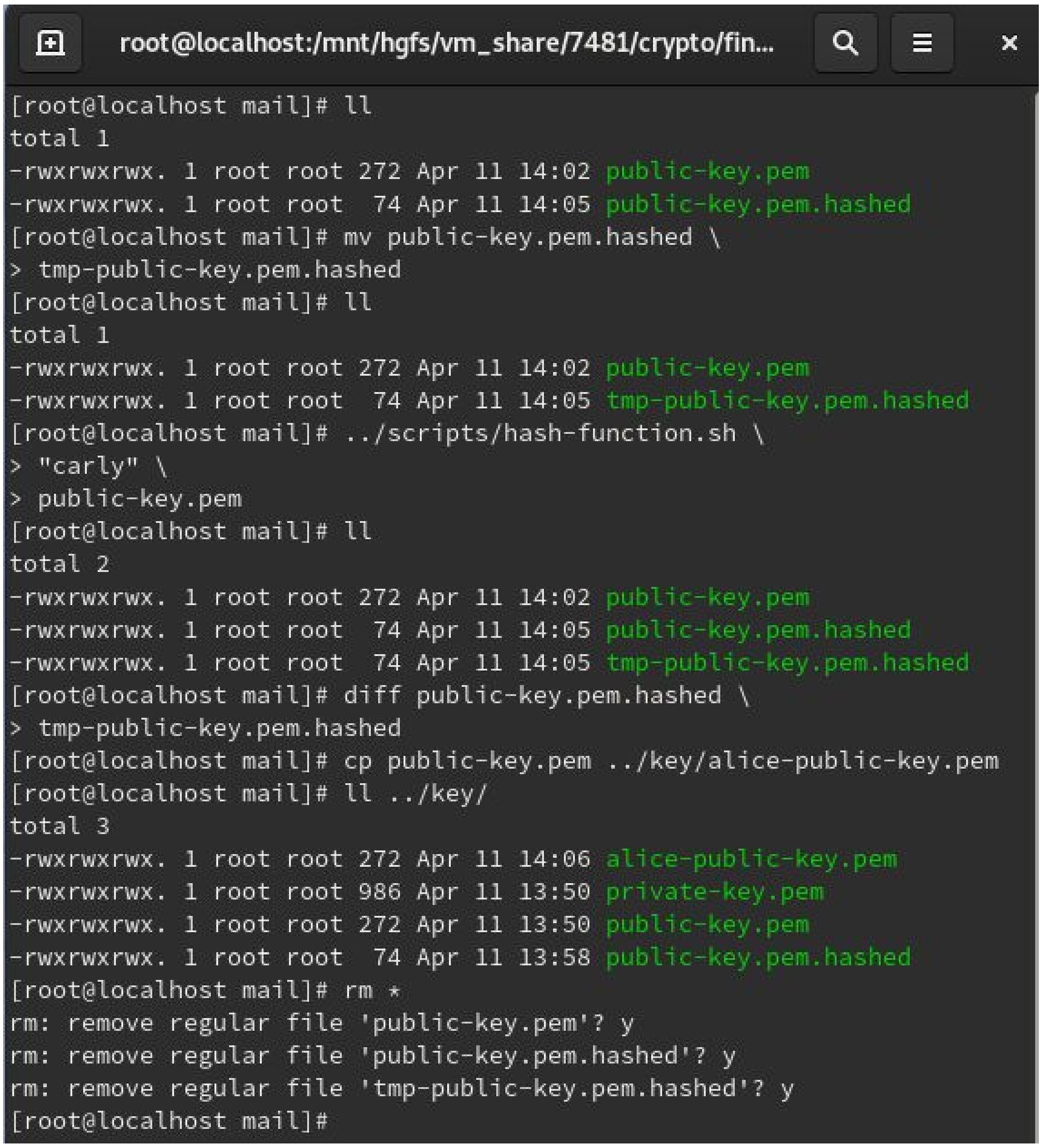


```
[root@localhost key]# ../scripts/hash-function.sh \
> "carly" \
> public-key.pem
[root@localhost key]# ll
total 2
-rwxrwxrwx. 1 root root 966 Apr 11 13:55 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:55 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:01 public-key.pem.hashed
```



```
[root@localhost Alice]# cp ./key/public-key.pem* ../Bob/mail
[root@localhost Alice]# ll ../Bob/mail/
total 1
-rwxrwxrwx. 1 root root 272 Apr 11 14:02 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:02 public-key.pem.hashed
```

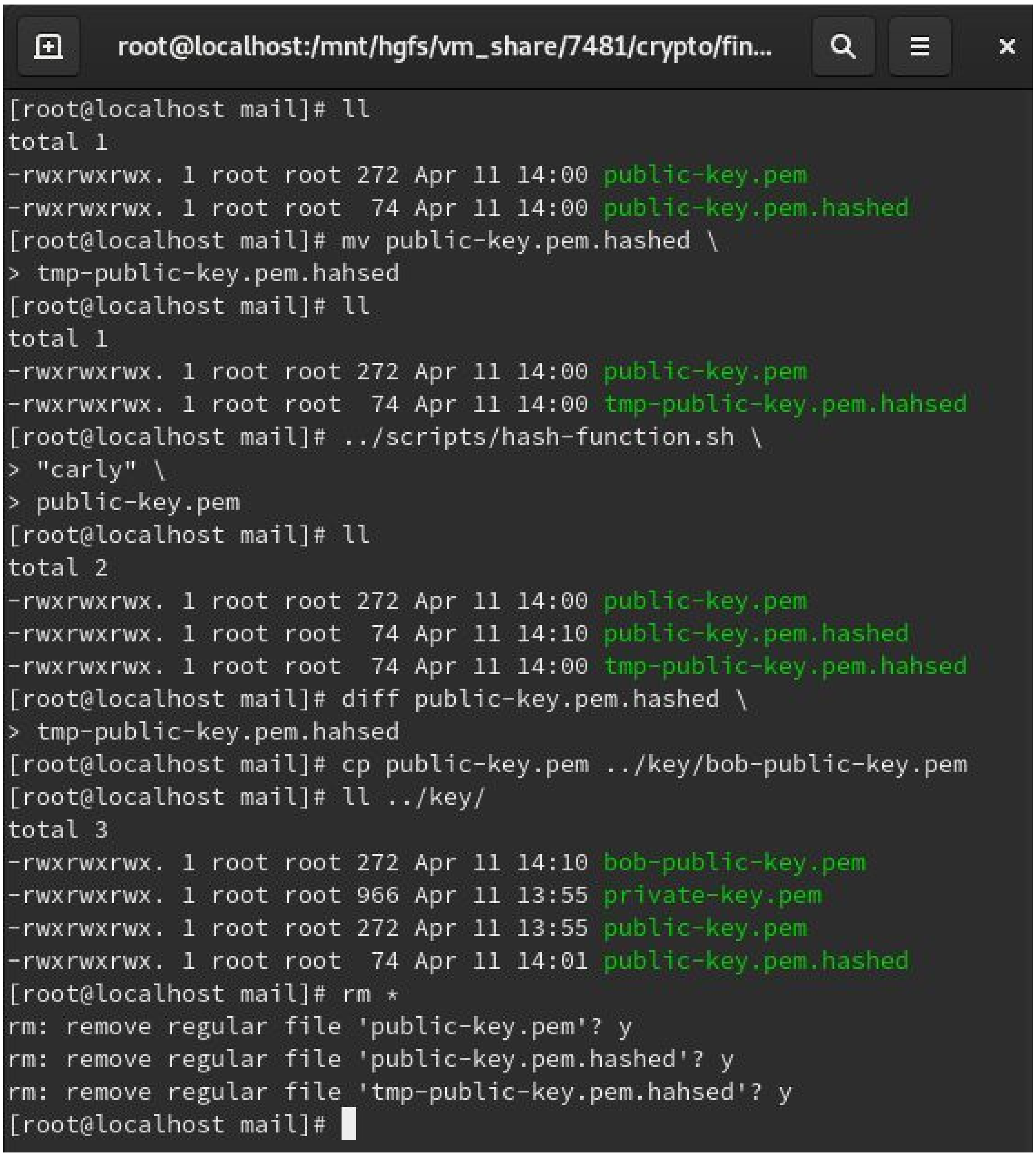
5) Bob retrieved both the original and hashed public key from his mailbox, and did a hash check to confirm that the public key is indeed from Alice. He then saved that public key, and cleared the mailbox.



root@localhost:/mnt/hgfs/vm_share/7481/crypto/fin...

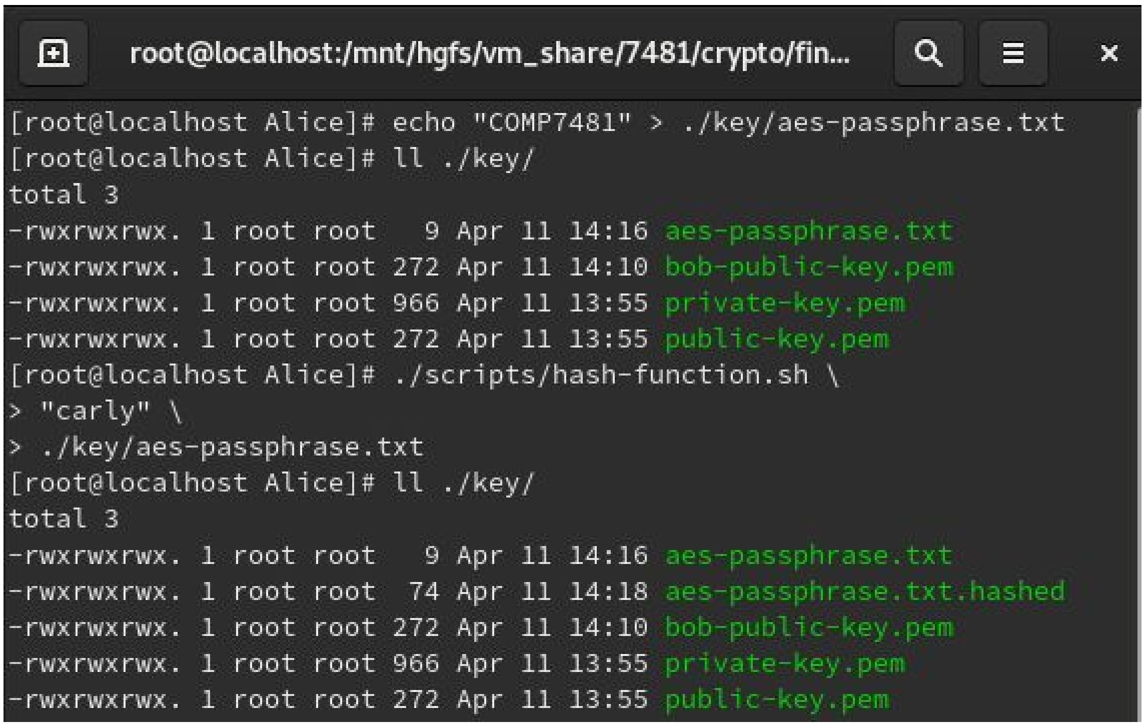
```
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 272 Apr 11 14:02 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:05 public-key.pem.hashed
[root@localhost mail]# mv public-key.pem.hashed \
> tmp-public-key.pem.hashed
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 272 Apr 11 14:02 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:05 tmp-public-key.pem.hashed
[root@localhost mail]# ../scripts/hash-function.sh \
> "carly" \
> public-key.pem
[root@localhost mail]# ll
total 2
-rwxrwxrwx. 1 root root 272 Apr 11 14:02 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:05 public-key.pem.hashed
-rwxrwxrwx. 1 root root  74 Apr 11 14:05 tmp-public-key.pem.hashed
[root@localhost mail]# diff public-key.pem.hashed \
> tmp-public-key.pem.hashed
[root@localhost mail]# cp public-key.pem ../key/alice-public-key.pem
[root@localhost mail]# ll ../key/
total 3
-rwxrwxrwx. 1 root root 272 Apr 11 14:06 alice-public-key.pem
-rwxrwxrwx. 1 root root 986 Apr 11 13:50 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:50 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 13:58 public-key.pem.hashed
[root@localhost mail]# rm *
rm: remove regular file 'public-key.pem'? y
rm: remove regular file 'public-key.pem.hashed'? y
rm: remove regular file 'tmp-public-key.pem.hashed'? y
[root@localhost mail]#
```

6) Alice retrieved both the original and hashed public key from her mailbox, and did a hash check to confirm that the public key is indeed from Bob. She then saved that public key, and cleared the mailbox.

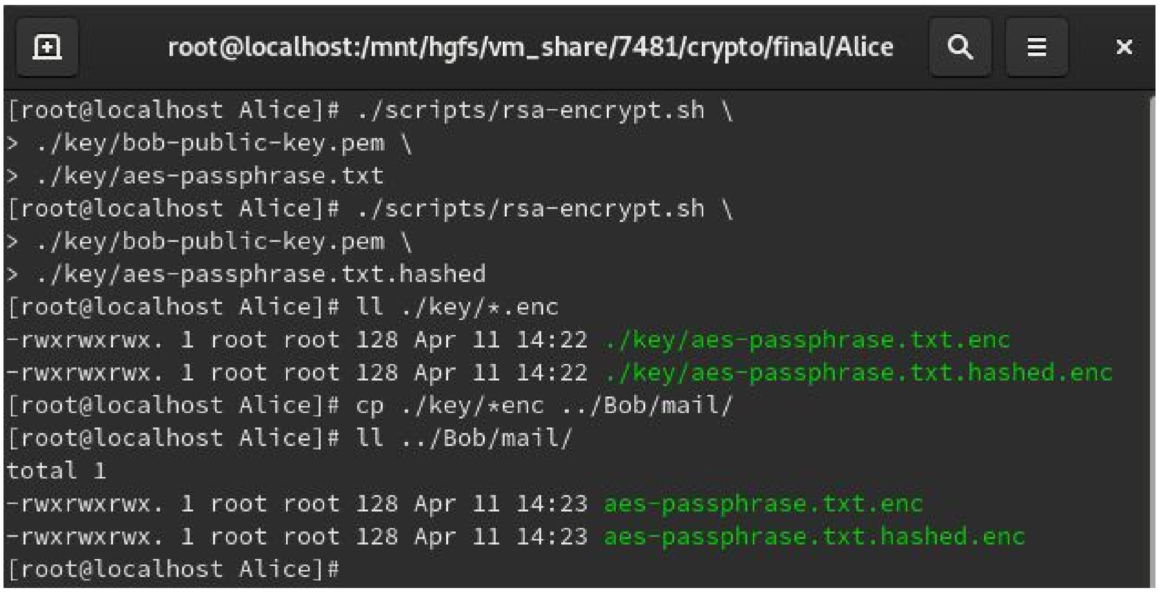


```
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 272 Apr 11 14:00 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:00 public-key.pem.hashed
[root@localhost mail]# mv public-key.pem.hashed \
> tmp-public-key.pem.hahsed
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 272 Apr 11 14:00 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:00 tmp-public-key.pem.hahsed
[root@localhost mail]# ../scripts/hash-function.sh \
> "carly" \
> public-key.pem
[root@localhost mail]# ll
total 2
-rwxrwxrwx. 1 root root 272 Apr 11 14:00 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:10 public-key.pem.hashed
-rwxrwxrwx. 1 root root  74 Apr 11 14:00 tmp-public-key.pem.hahsed
[root@localhost mail]# diff public-key.pem.hashed \
> tmp-public-key.pem.hahsed
[root@localhost mail]# cp public-key.pem ../key/bob-public-key.pem
[root@localhost mail]# ll ../key/
total 3
-rwxrwxrwx. 1 root root 272 Apr 11 14:10 bob-public-key.pem
-rwxrwxrwx. 1 root root 966 Apr 11 13:55 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:55 public-key.pem
-rwxrwxrwx. 1 root root  74 Apr 11 14:01 public-key.pem.hashed
[root@localhost mail]# rm *
rm: remove regular file 'public-key.pem'? y
rm: remove regular file 'public-key.pem.hashed'? y
rm: remove regular file 'tmp-public-key.pem.hahsed'? y
[root@localhost mail]# 
```

7) Alice decided to use the passphrase "**COMP7481**" for AES encryption. To securely pass this passphrase to Bob, she hash the passphrase with hash key "**carly**", encrypt both the original and hashed passphrase with Bob's public key, and mail all to Bob

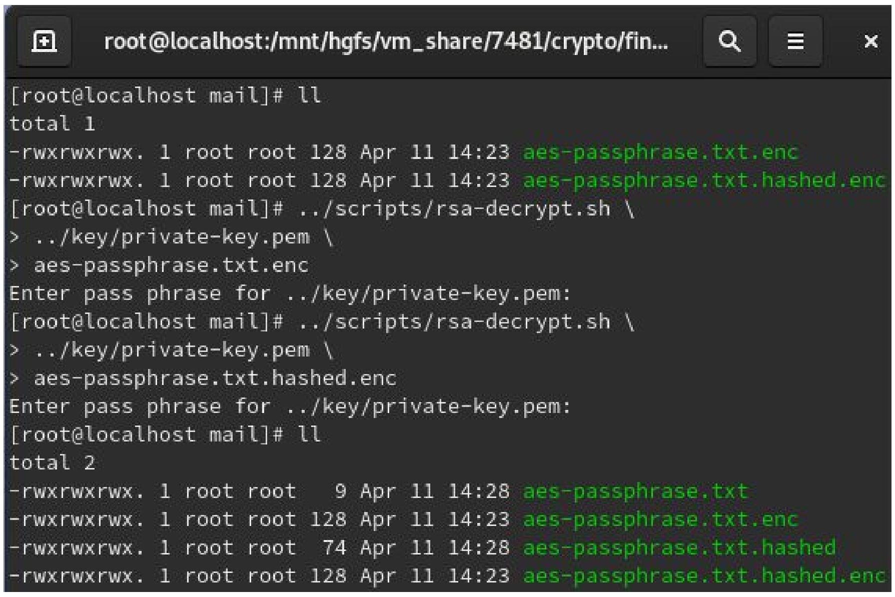


```
[root@localhost Alice]# echo "COMP7481" > ./key/aes-passphrase.txt
[root@localhost Alice]# ll ./key/
total 3
-rwxrwxrwx. 1 root root   9 Apr 11 14:16 aes-passphrase.txt
-rwxrwxrwx. 1 root root 272 Apr 11 14:10 bob-public-key.pem
-rwxrwxrwx. 1 root root 966 Apr 11 13:55 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:55 public-key.pem
[root@localhost Alice]# ./scripts/hash-function.sh \
> "carly" \
> ./key/aes-passphrase.txt
[root@localhost Alice]# ll ./key/
total 3
-rwxrwxrwx. 1 root root   9 Apr 11 14:16 aes-passphrase.txt
-rwxrwxrwx. 1 root root  74 Apr 11 14:18 aes-passphrase.txt.hashed
-rwxrwxrwx. 1 root root 272 Apr 11 14:10 bob-public-key.pem
-rwxrwxrwx. 1 root root 966 Apr 11 13:55 private-key.pem
-rwxrwxrwx. 1 root root 272 Apr 11 13:55 public-key.pem
```



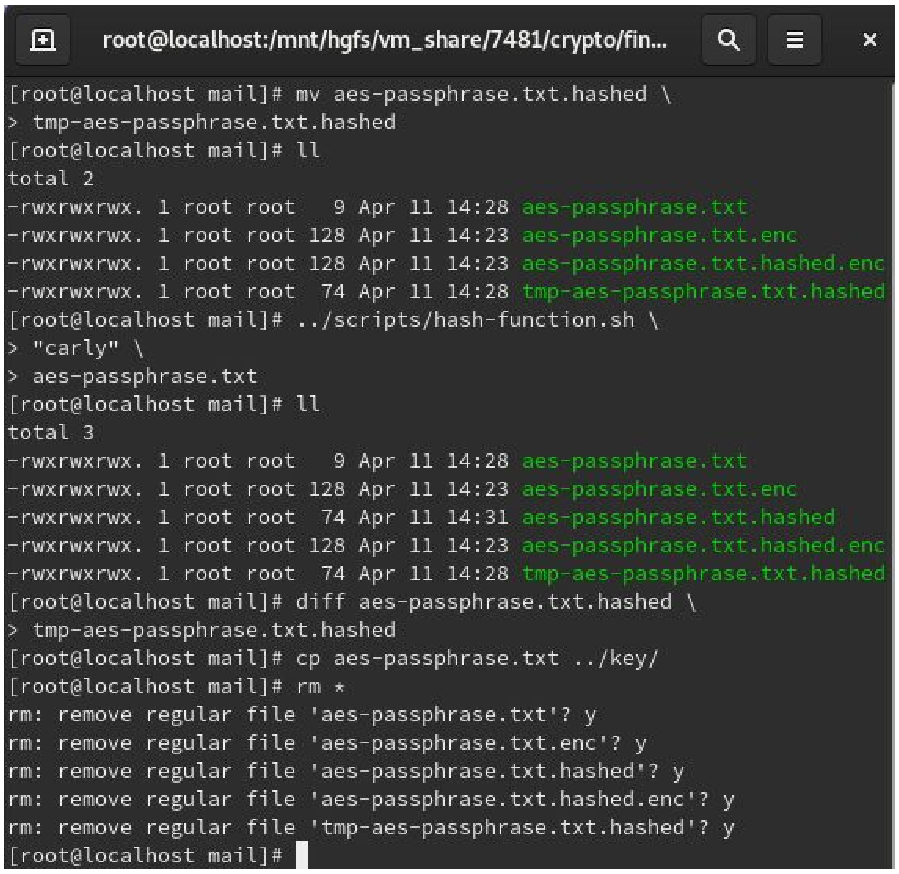
```
[root@localhost Alice]# ./scripts/rsa-encrypt.sh \
> ./key/bob-public-key.pem \
> ./key/aes-passphrase.txt
[root@localhost Alice]# ./scripts/rsa-encrypt.sh \
> ./key/bob-public-key.pem \
> ./key/aes-passphrase.txt.hashed
[root@localhost Alice]# ll ./key/*.enc
-rwxrwxrwx. 1 root root 128 Apr 11 14:22 ./key/aes-passphrase.txt.enc
-rwxrwxrwx. 1 root root 128 Apr 11 14:22 ./key/aes-passphrase.txt.hashed.enc
[root@localhost Alice]# cp ./key/*enc ../Bob/mail/
[root@localhost Alice]# ll ../Bob/mail/
total 1
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.enc
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.hashed.enc
[root@localhost Alice]#
```

8) Bob retrieved both the original and hashed AES key from his mailbox. He first decrypted both files with his private RSA key. Next, he did a hash check to confirm that the AES key is indeed from Alice. He then saved that AES key, and cleared the mailbox.



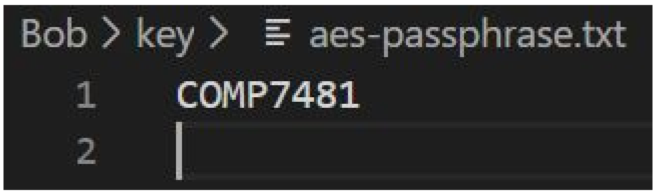
root@localhost:/mnt/hgfs/vm_share/7481/crypto/fin...

```
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.enc
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.hashed.enc
[root@localhost mail]# ../scripts/rsa-decrypt.sh \
> ../key/private-key.pem \
> aes-passphrase.txt.enc
Enter pass phrase for ../key/private-key.pem:
[root@localhost mail]# ../scripts/rsa-decrypt.sh \
> ../key/private-key.pem \
> aes-passphrase.txt.hashed.enc
Enter pass phrase for ../key/private-key.pem:
[root@localhost mail]# ll
total 2
-rwxrwxrwx. 1 root root   9 Apr 11 14:28 aes-passphrase.txt
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.enc
-rwxrwxrwx. 1 root root  74 Apr 11 14:28 aes-passphrase.txt.hashed
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.hashed.enc
```



root@localhost:/mnt/hgfs/vm_share/7481/crypto/fin...

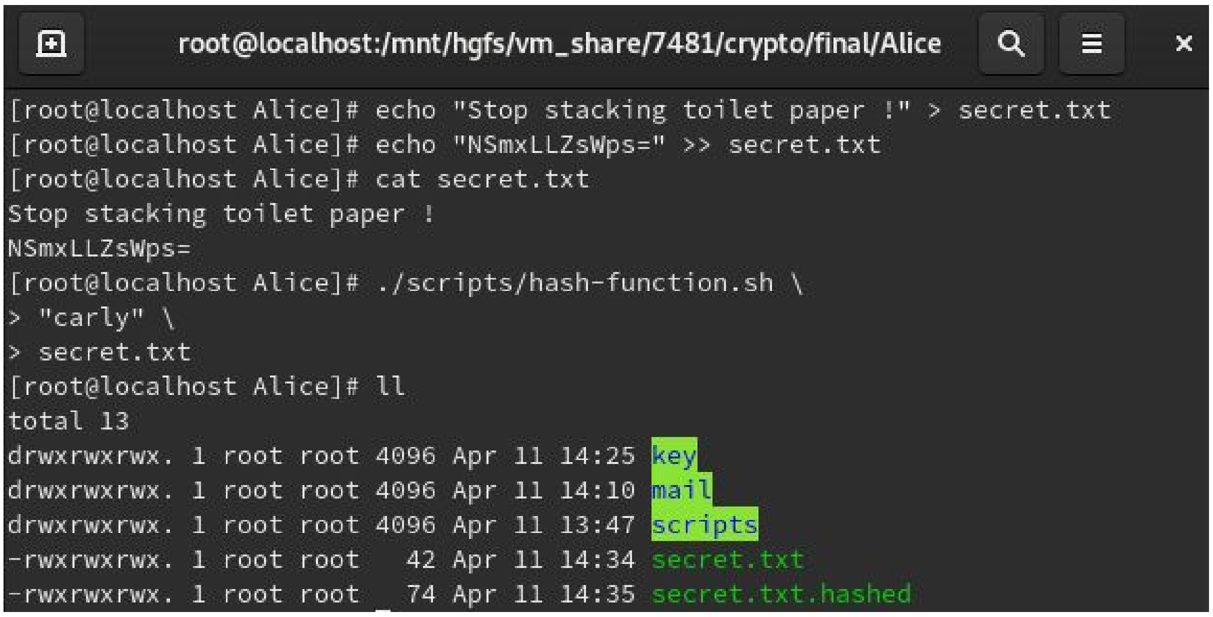
```
[root@localhost mail]# mv aes-passphrase.txt.hashed \
> tmp-aes-passphrase.txt.hashed
[root@localhost mail]# ll
total 2
-rwxrwxrwx. 1 root root   9 Apr 11 14:28 aes-passphrase.txt
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.enc
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.hashed.enc
-rwxrwxrwx. 1 root root  74 Apr 11 14:28 tmp-aes-passphrase.txt.hashed
[root@localhost mail]# ../scripts/hash-function.sh \
> "carly" \
> aes-passphrase.txt
[root@localhost mail]# ll
total 3
-rwxrwxrwx. 1 root root   9 Apr 11 14:28 aes-passphrase.txt
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.enc
-rwxrwxrwx. 1 root root  74 Apr 11 14:31 aes-passphrase.txt.hashed
-rwxrwxrwx. 1 root root 128 Apr 11 14:23 aes-passphrase.txt.hashed.enc
-rwxrwxrwx. 1 root root  74 Apr 11 14:28 tmp-aes-passphrase.txt.hashed
[root@localhost mail]# diff aes-passphrase.txt.hashed \
> tmp-aes-passphrase.txt.hashed
[root@localhost mail]# cp aes-passphrase.txt ../key/
[root@localhost mail]# rm *
rm: remove regular file 'aes-passphrase.txt'? y
rm: remove regular file 'aes-passphrase.txt.enc'? y
rm: remove regular file 'aes-passphrase.txt.hashed'? y
rm: remove regular file 'aes-passphrase.txt.hashed.enc'? y
rm: remove regular file 'tmp-aes-passphrase.txt.hashed'? y
[root@localhost mail]#
```



Bob > key > aes-passphrase.txt

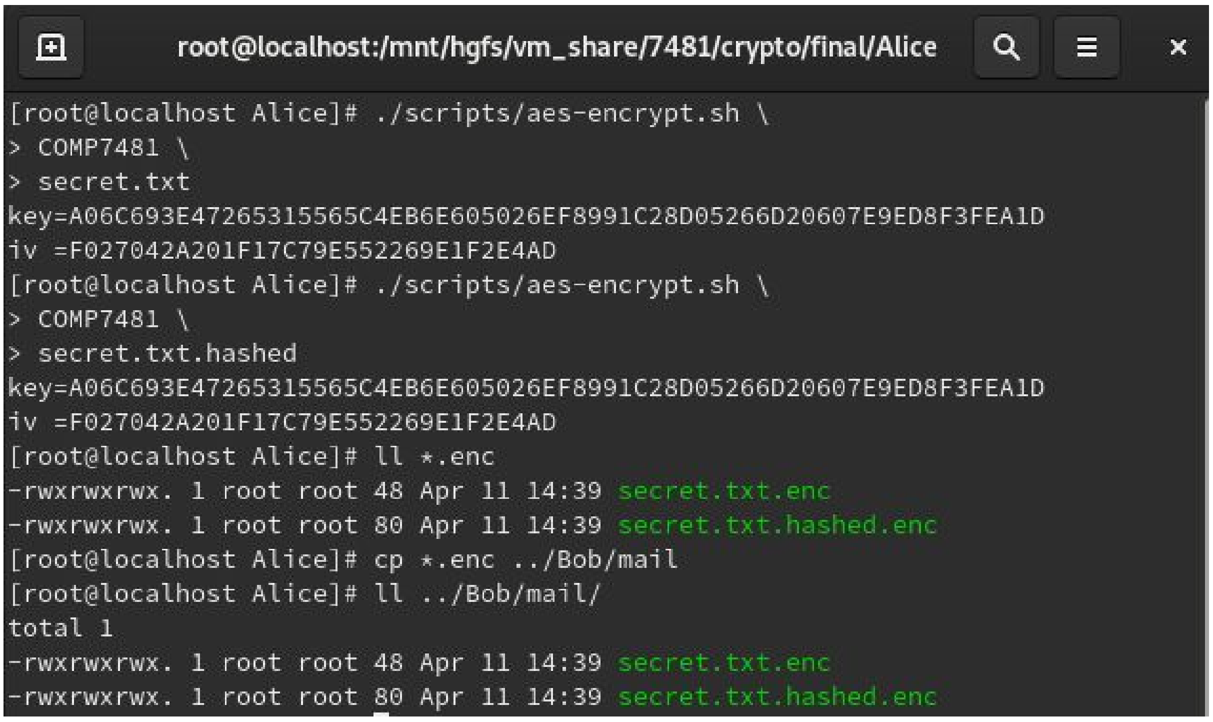
```
1    COMP7481
2
```

9) Bob let Alice know he had successfully retrieved the AES key. (a simple echo would do). Next, Alice created a secret file named “secret.txt”. She appended the hash salt to the text file and hash the file with a hash key. After, she encrypted both the original and hashed file with AES using passphrase “COMP7481”, and directed both files to Bob’s mailbox.



root@localhost:/mnt/hgfs/vm_share/7481/crypto/final/Alice

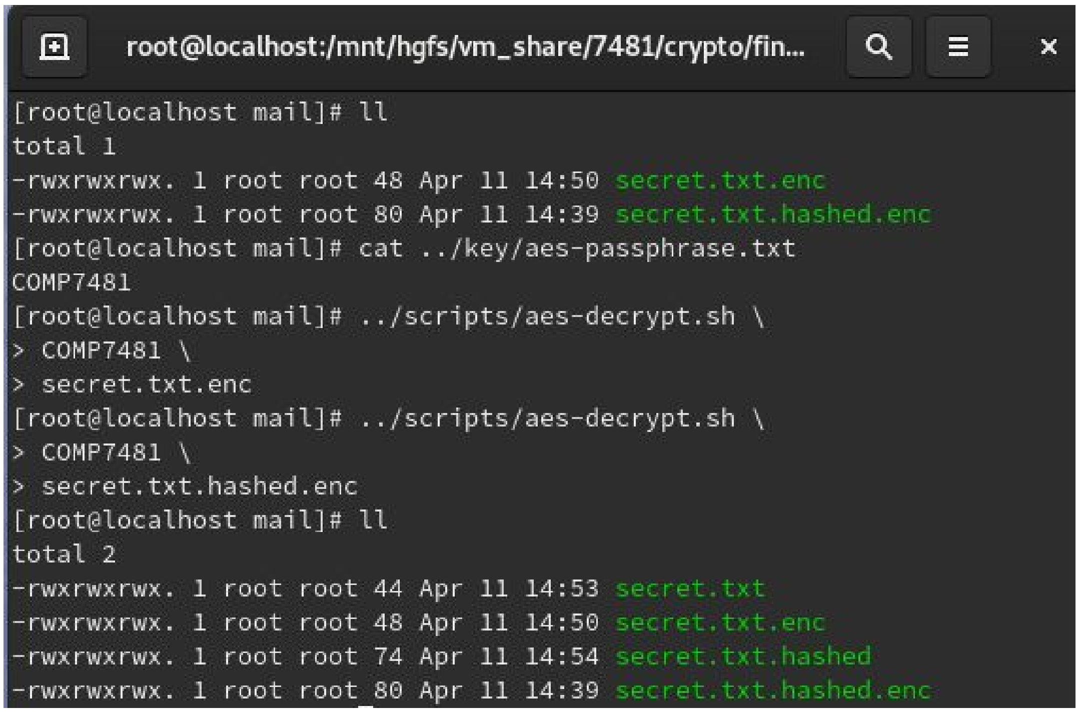
```
[root@localhost Alice]# echo "Stop stacking toilet paper !" > secret.txt
[root@localhost Alice]# echo "NSmxLLZsWps=" >> secret.txt
[root@localhost Alice]# cat secret.txt
Stop stacking toilet paper !
NSmxLLZsWps=
[root@localhost Alice]# ./scripts/hash-function.sh \
> "carly" \
> secret.txt
[root@localhost Alice]# ll
total 13
drwxrwxrwx. 1 root root 4096 Apr 11 14:25 key
drwxrwxrwx. 1 root root 4096 Apr 11 14:10 mail
drwxrwxrwx. 1 root root 4096 Apr 11 13:47 scripts
-rwxrwxrwx. 1 root root   42 Apr 11 14:34 secret.txt
-rwxrwxrwx. 1 root root   74 Apr 11 14:35 secret.txt.hashed
```



root@localhost:/mnt/hgfs/vm_share/7481/crypto/final/Alice

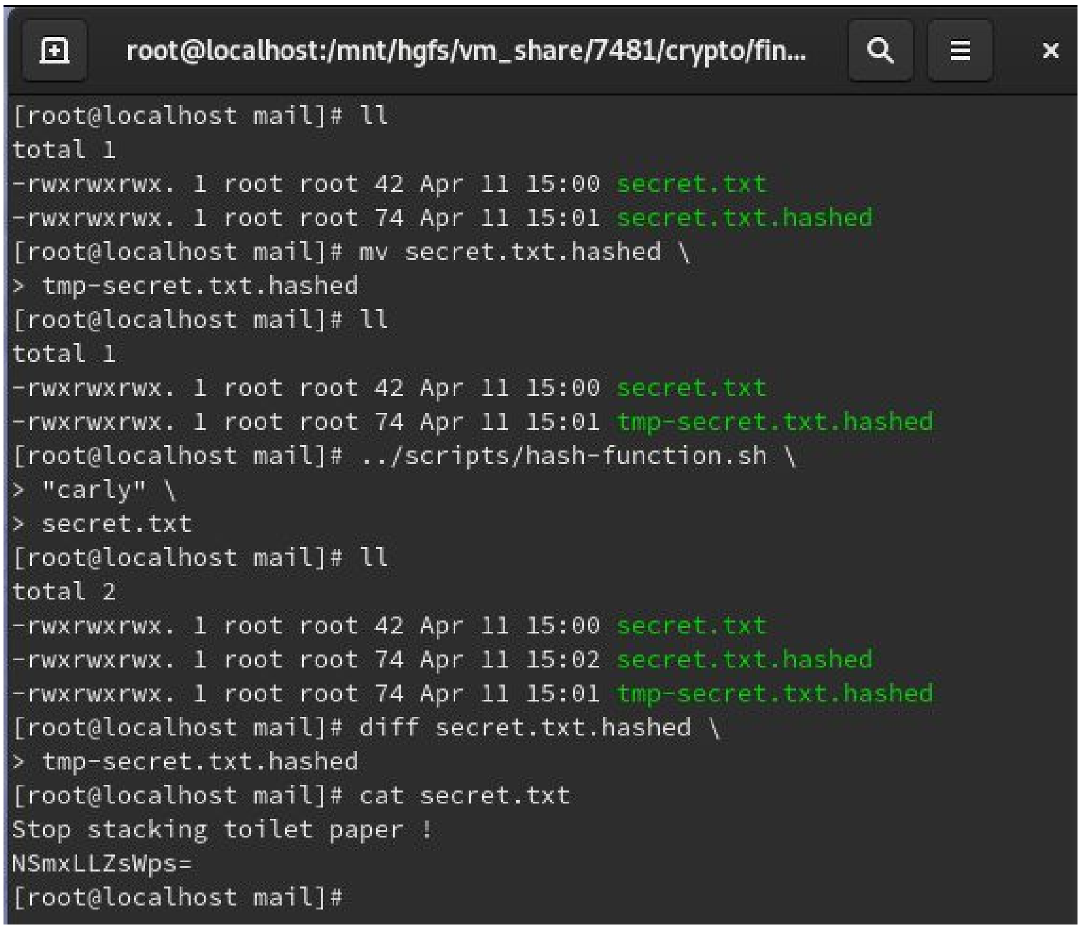
```
[root@localhost Alice]# ./scripts/aes-encrypt.sh \
> COMP7481 \
> secret.txt
key=A06C693E47265315565C4EB6E605026EF8991C28D05266D20607E9ED8F3FEA1D
iv =F027042A201F17C79E552269E1F2E4AD
[root@localhost Alice]# ./scripts/aes-encrypt.sh \
> COMP7481 \
> secret.txt.hashed
key=A06C693E47265315565C4EB6E605026EF8991C28D05266D20607E9ED8F3FEA1D
iv =F027042A201F17C79E552269E1F2E4AD
[root@localhost Alice]# ll *.enc
-rwxrwxrwx. 1 root root 48 Apr 11 14:39 secret.txt.enc
-rwxrwxrwx. 1 root root 80 Apr 11 14:39 secret.txt.hashed.enc
[root@localhost Alice]# cp *.enc ../Bob/mail
[root@localhost Alice]# ll ../Bob/mail/
total 1
-rwxrwxrwx. 1 root root 48 Apr 11 14:39 secret.txt.enc
-rwxrwxrwx. 1 root root 80 Apr 11 14:39 secret.txt.hashed.enc
```

10) Bob retrieved the encrypted files from mail. He first decrypted both .enc files with AES using the passphrase received earlier (COMP7481). To confirm that the received secret message has not been altered, he checked the salt and hash. The checking passed.
Bob had successfully receive Alice's secret message



root@localhost:/mnt/hgfs/vm_share/7481/crypto/fin...

```
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 48 Apr 11 14:50 secret.txt.enc
-rwxrwxrwx. 1 root root 80 Apr 11 14:39 secret.txt.hashed.enc
[root@localhost mail]# cat ../key/aes-passphrase.txt
COMP7481
[root@localhost mail]# ../scripts/aes-decrypt.sh \
> COMP7481 \
> secret.txt.enc
[root@localhost mail]# ../scripts/aes-decrypt.sh \
> COMP7481 \
> secret.txt.hashed.enc
[root@localhost mail]# ll
total 2
-rwxrwxrwx. 1 root root 44 Apr 11 14:53 secret.txt
-rwxrwxrwx. 1 root root 48 Apr 11 14:50 secret.txt.enc
-rwxrwxrwx. 1 root root 74 Apr 11 14:54 secret.txt.hashed
-rwxrwxrwx. 1 root root 80 Apr 11 14:39 secret.txt.hashed.enc
```



root@localhost:/mnt/hgfs/vm_share/7481/crypto/fin...

```
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 42 Apr 11 15:00 secret.txt
-rwxrwxrwx. 1 root root 74 Apr 11 15:01 secret.txt.hashed
[root@localhost mail]# mv secret.txt.hashed \
> tmp-secret.txt.hashed
[root@localhost mail]# ll
total 1
-rwxrwxrwx. 1 root root 42 Apr 11 15:00 secret.txt
-rwxrwxrwx. 1 root root 74 Apr 11 15:01 tmp-secret.txt.hashed
[root@localhost mail]# ../scripts/hash-function.sh \
> "carly" \
> secret.txt
[root@localhost mail]# ll
total 2
-rwxrwxrwx. 1 root root 42 Apr 11 15:00 secret.txt
-rwxrwxrwx. 1 root root 74 Apr 11 15:02 secret.txt.hashed
-rwxrwxrwx. 1 root root 74 Apr 11 15:01 tmp-secret.txt.hashed
[root@localhost mail]# diff secret.txt.hashed \
> tmp-secret.txt.hashed
[root@localhost mail]# cat secret.txt
Stop stacking toilet paper !
NSmxLLZsWps=
[root@localhost mail]#
```

Conclusion

The system seems fairly robust in terms of security. The RSA passphrase provides an extra layer of protection. For instance, if a third party (Alex) somehow gains access to Bob's private key, he would have to know the passphrase in order to decrypt the message. The RSA keys are only used for AES key exchange, this decreases the cost for encryption while keeping the security strength at an acceptable level.

To increase the security strength of this system even more, we could choose to use dynamic salt (different salt for every encryption process), different hash algorithm for both key exchange and data encryption, and so on. However, for the scope of this experiment, this system is more than efficient for simple private message exchange.