

1.

a)

Alice needs to use her private key to encrypt the symmetric key. Alice can start by generating a random symmetric key to be used for transmission. She can use this key to encrypt the data file. She can use Bob's public key to encrypt the symmetric key using asymmetric cryptography. She would then proceed to combine the encrypted symmetric key with the encrypted data file into a message to transmit.

b)

Bob can start by decrypting the encrypted symmetric key with use of his private key to give him the original symmetric key, and Bob needs to use Alice's public key to decrypt the symmetric key. He can then get the original data file using the symmetric key to decrypt the encrypted data file.

2.

a)

```
phishingWebsite [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminal

cipher1.txt      Music          Pictures
Desktop          openssl-1.0.1  plain.txt
Documents        openssl_1.0.1-4ubuntu5.11.debian.tar.gz Public
Downloads        openssl_1.0.1-4ubuntu5.11.dsc    Templates
elggData         openssl_1.0.1.orig.tar.gz      Videos
examples.desktop phishSite

[11/07/2023 12:34] seed@ubuntu:~$ vim plain.txt
[11/07/2023 12:36] seed@ubuntu:~$ openssl enc -aes-256-cbc -e -in plain.txt -out
cipher1.txt -K 4673a788a9c292a5ba3099c2fc1e4fb0 -iv e9a11a81cddb6b7f370d94015ea
cd72d
[11/07/2023 12:37] seed@ubuntu:~$ xxd -p cipher1.txt
8fc3915a02b70b176377b7907784c10668e06065566dfffad84c520fa13e
12160923722c1a75f364dc2a31c0f3dad9d594a327332d8b1356ff504130
8d6d4247
[11/07/2023 12:37] seed@ubuntu:~$ openssl enc -aes-256-cbc -e -in plain.txt -out
cipher2.txt -K 4673a788a9c292a5ba3099c2fc1e4fb0 -iv e9a11a81cddb6b7f370d94015ea
cd72e
[11/07/2023 12:38] seed@ubuntu:~$ openssl enc -aes-256-cbc -d -in cipher2.txt -o
ut plain2.txt -K 4673a788a9c292a5ba3099c2fc1e4fb0 -iv e9a11a81cddb6b7f370d94015e
acd72e
[11/07/2023 12:38] seed@ubuntu:~$ cat plain2.txt
The quick brown fox jumps over the lazy dog - Austin
[11/07/2023 12:38] seed@ubuntu:~$
```

b)

```
[11/07/2023 12:38] seed@ubuntu:~$ openssl rand -hex 32 > hmacKey.txt
[11/07/2023 12:46] seed@ubuntu:~$ openssl dgst -hmac "$(cat hmacKey.txt)" -sha256
6 -binary plain.txt > tag.bin
[11/07/2023 12:48] seed@ubuntu:~$ openssl dgst -hex tag.bin
MD5(tag.bin)= 9581c654731cad569f6bf977fceaaceda
[11/07/2023 12:48] seed@ubuntu:~$ xxd -p cipher1.txt | openssl dgst -sha256 -hma
c 10101010101010101010101010101010
(stdin)= 87a69bf589e73dcaa1a17b09b5374a780e8ceb43ce60e5da8674c86f9a599d0e
[11/07/2023 12:51] seed@ubuntu:~$
```

3.

Bob can use public key cryptography to confirm that Alice is Alice. First, Bob can make a message for Alice to respond to. Bob would then encrypt the message using Alice's public key and send the encrypted message to Alice. Alice would then decrypt the message using her private key. Alice would send the

decrypted message back to Bob, who would compare her decrypted message with his original message. If they're the same, Alice is indeed Alice.