5.

Download the public key from the public key-server.

To encrypt a message that another person can decrypt, we must have their public key.

Import the downloaded public key to gpg keys.

```
# gpg --import 34FB0BA2155A9D20D248BD0E7D11327265445CF2.asc
```

```
bbr@EME17-G7064PKR:~/Documents/WSO2/2_Information_Security_Concepts/Answers/10$ gpg --import 34FB0BA2155A9D20D248BD0E7D1
1327265445CF2.asc
gpg: key 7D11327265445CF2: public key "Ayoma Wijethunga <ayomawdb@gmail.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

The key is imported, and you are shown the name and email address associated with that key. Obviously, that should match the person you received it from.

The **--encrypt** option tells gpg to encrypt the file, and the **--armor** option tells gpg to create an ASCII file. The **-r** (recipient) option must be followed by the email address of the person you're sending the file to.

```
# gpg --output nistspecialpublication800-100.pdf.gpg --encrypt --recipient
ayomawdb@gmail.com nistspecialpublication800-100.pdf
```

```
bbr@EME17-G7064PKR:~/Documents/WSO2/2_Information_Security_Concepts/Answers/10$ gpg --output
nistspecialpublication800-100.pdf.gpg --encrypt --recipient ayomawdb@gmail.com nistspecialpub
lication800-100.pdf
gpg: 32F2268D88105937: There is no assurance this key belongs to the named user

sub  rsa4096/32F2268D88105937 2021-07-05 Ayoma Wijethunga <ayomawdb@gmail.com>
 Primary key fingerprint: 34FB 0BA2 155A 9D20 D248  BD0E 7D11 3272 6544 5CF2
      Subkey fingerprint: 0A1B DD38 0AA0 0E70 B440  1B90 32F2 268D 8810 5937

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```