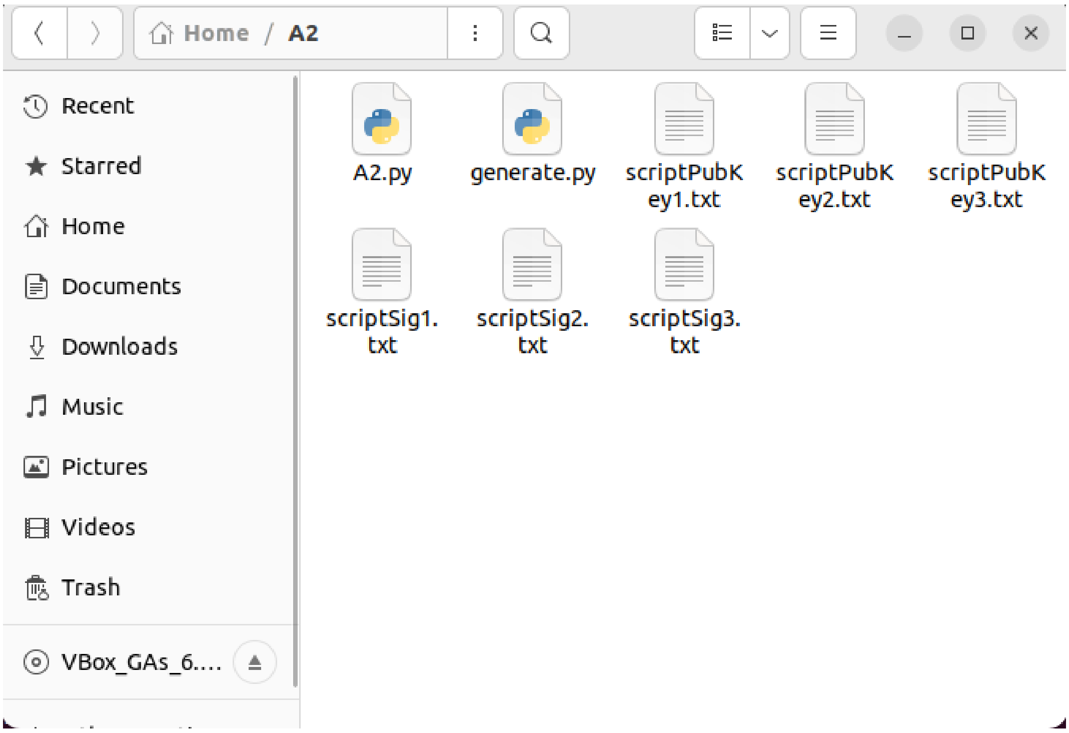
Bronson Chan

CSCI301 Assignment 2 Report

These are all the files in the folder at first.



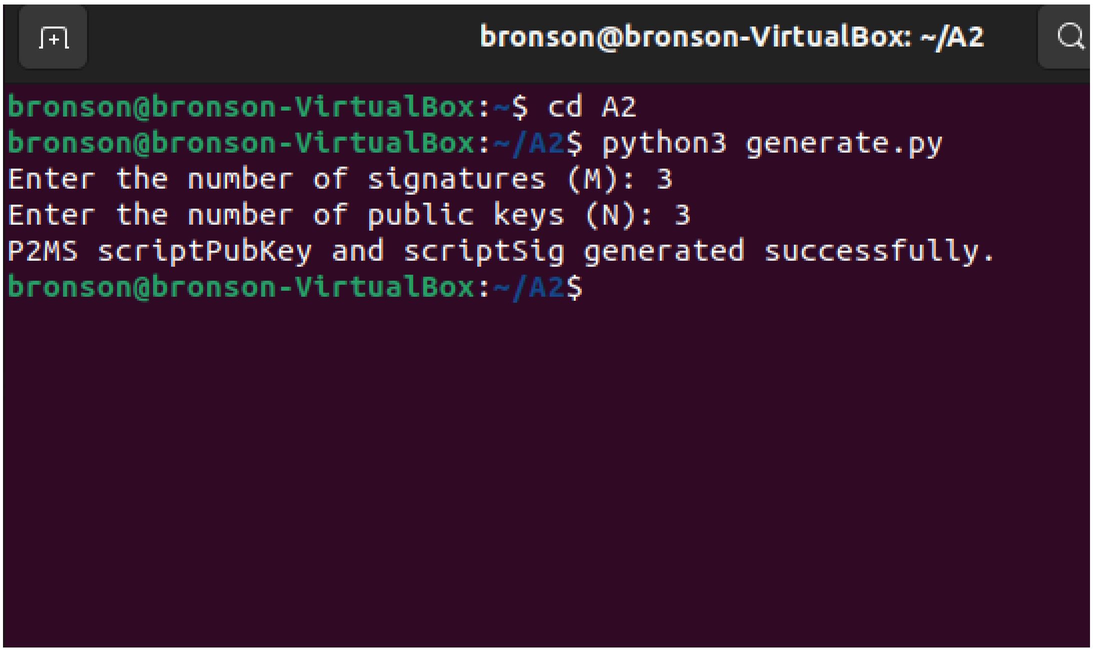
A2.py
generate.py
scriptPubKey1.txt
scriptPubKey2.txt
scriptPubKey3.txt
scriptSig1.txt
scriptSig2.txt
scriptSig3.txt

generate.py (program which generate user input M and N (number of signatures and public key) and store them to the respective textfiles. Eg, scriptPubKey1.txt and scriptSig1.txt

a2.py (program which reads from the respective files and verifies the value)

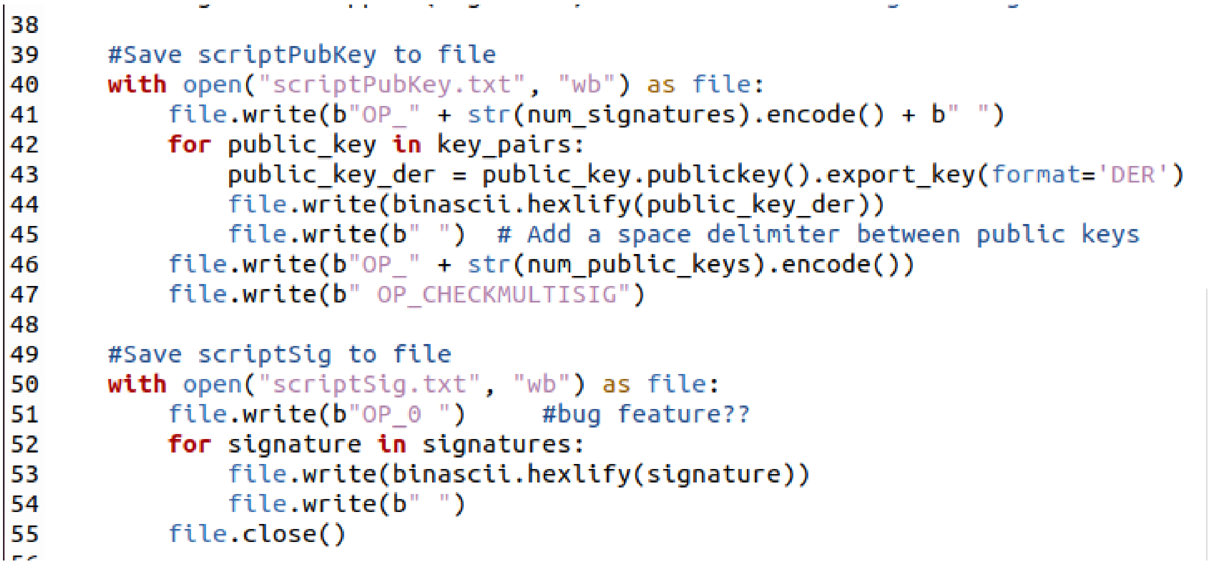
Running generate.py

Generation of keys and signatures (scriptPubKey & scriptSig) which had already been done in the text files 1 2 and 3



```
bronson@bronson-VirtualBox:~$ cd A2
bronson@bronson-VirtualBox:~/A2$ python3 generate.py
Enter the number of signatures (M): 3
Enter the number of public keys (N): 3
P2MS scriptPubKey and scriptSig generated successfully.
bronson@bronson-VirtualBox:~/A2$
```

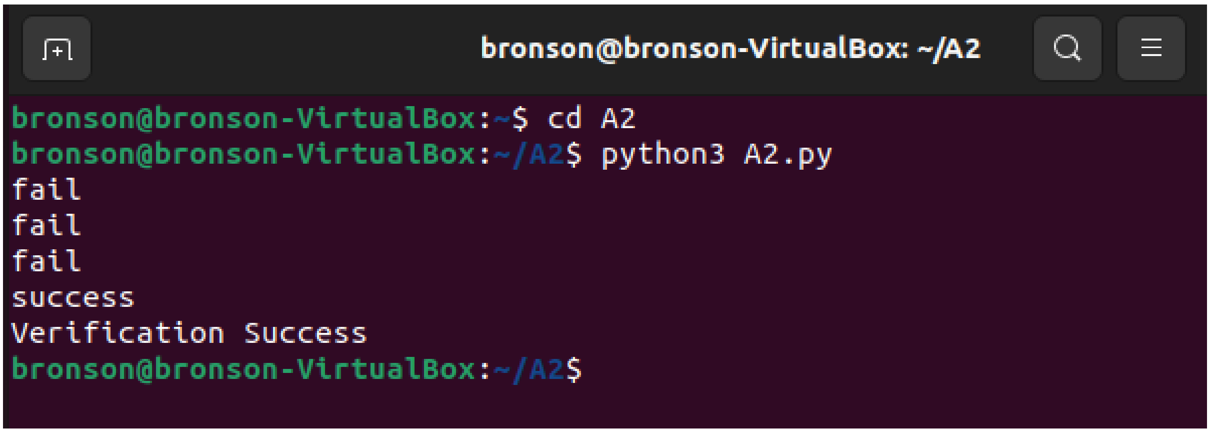
Running generate.py will create new textfiles and save them as the format of scriptPubKey and scriptSig



```
38
39      #Save scriptPubKey to file
40      with open("scriptPubKey.txt", "wb") as file:
41          file.write(b"OP_" + str(num_signatures).encode() + b" ")
42          for public_key in key_pairs:
43              public_key_der = public_key.publickey().export_key(format='DER')
44              file.write(binascii.hexlify(public_key_der))
45              file.write(b" ")  # Add a space delimiter between public keys
46          file.write(b"OP_" + str(num_public_keys).encode())
47          file.write(b" OP_CHECKMULTISIG")
48
49      #Save scriptSig to file
50      with open("scriptSig.txt", "wb") as file:
51          file.write(b"OP_0 ")     #bug feature??
52          for signature in signatures:
53              file.write(binascii.hexlify(signature))
54              file.write(b" ")
55          file.close()
```

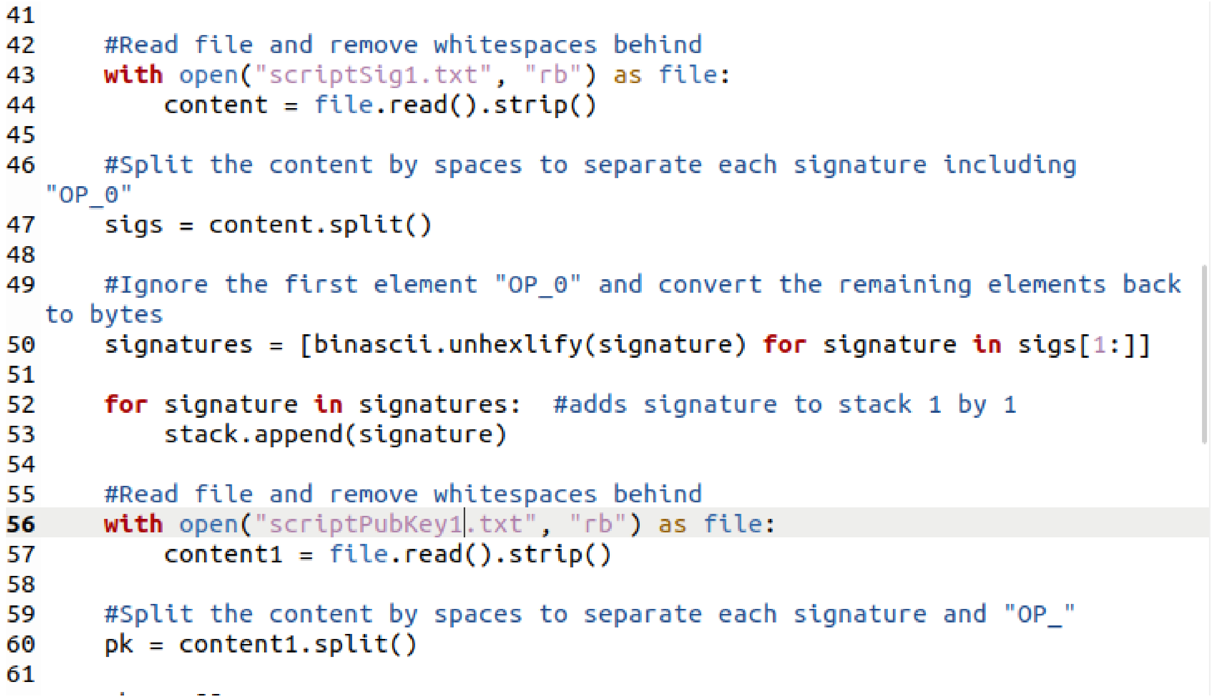
Hardcode the file names at line 40 and line 50 before generating, in this case which is scriptPubKey.txt and scriptSig.txt (which I had deleted from the folder as I already have 3 sets of scriptPubKey.txt and scriptSig.txt)

Running of A2.py



```
bronson@bronson-VirtualBox:~$ cd A2
bronson@bronson-VirtualBox:~/A2$ python3 A2.py
fail
fail
fail
success
Verification Success
bronson@bronson-VirtualBox:~/A2$
```

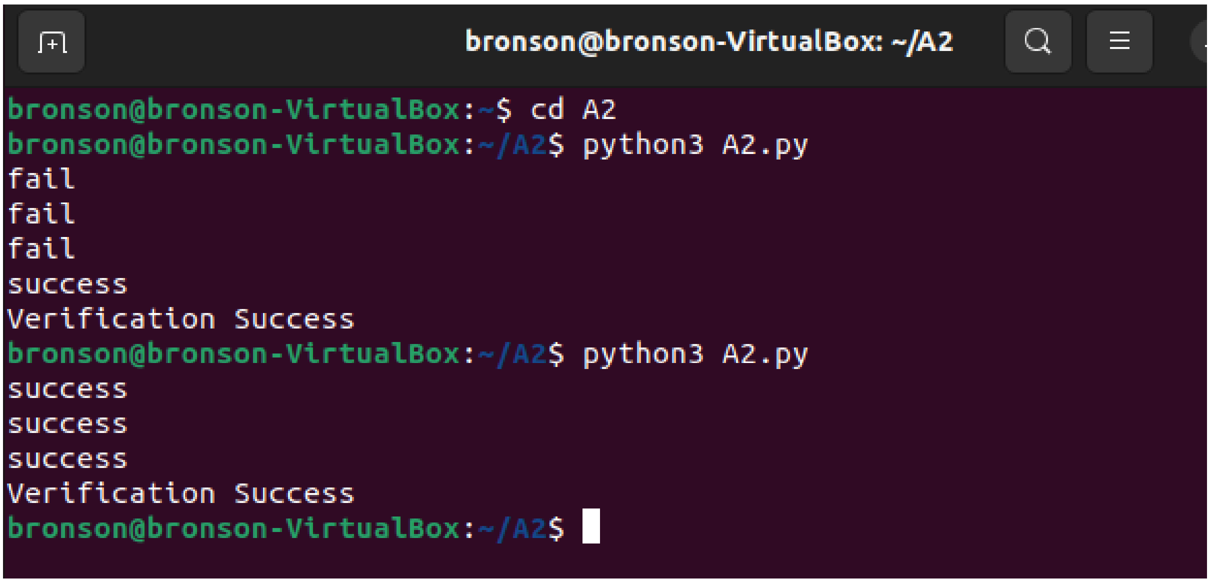
I have programmed to print("fail") if the public key and signature doesn't verify and print("success") if it does verify in the for loop. But in the end once all is verified, if True is left in the stack, the program will print("Verification Success") else if the verification went wrong, False will be left in the stack and the program will print("Verification Fail")



```
41
42      #Read file and remove whitespaces behind
43      with open("scriptSig1.txt", "rb") as file:
44          content = file.read().strip()
45
46      #Split the content by spaces to separate each signature including
    "OP_0"
47      sigs = content.split()
48
49      #Ignore the first element "OP_0" and convert the remaining elements back
    to bytes
50      signatures = [binascii.unhexlify(signature) for signature in sigs[1:]]
51
52      for signature in signatures:  #adds signature to stack 1 by 1
53          stack.append(signature)
54
55      #Read file and remove whitespaces behind
56      with open("scriptPubKey1.txt", "rb") as file:
57          content1 = file.read().strip()
58
59      #Split the content by spaces to separate each signature and "OP_"
60      pk = content1.split()
61
```

In line 43 and line 56, remember to hardcode the correct files to run. scriptPubKey1 must run with scriptSig1 and scriptPubKey2 must run with scriptSig2 else the program will print(“Verification Fail”)

For now, I will put it as scriptSig.txt and scriptPubKey.txt so that can be used immediately after running generate.py



```
bronson@bronson-VirtualBox:~$ cd A2
bronson@bronson-VirtualBox:~/A2$ python3 A2.py
fail
fail
fail
success
Verification Success
bronson@bronson-VirtualBox:~/A2$ python3 A2.py
success
success
success
Verification Success
bronson@bronson-VirtualBox:~/A2$ 
```

The second try will not show any print(“fail”) as there are 3 signatures and 3 public keys.