Sean Stephen (UOW ID 7311230) Report For CSCI301 Assignment 2

List of Files:

Creating/Executing Program: a2_generate.py / a2_verify.py

Output Files: scriptPubKey.txt, scriptSig.txt

Three different pairs of scriptSig are stored in scriptSig.txt. Four different pairs of scriptPubKey containing 3 matching pairs to the signatures are stored in scriptPubKey.txt

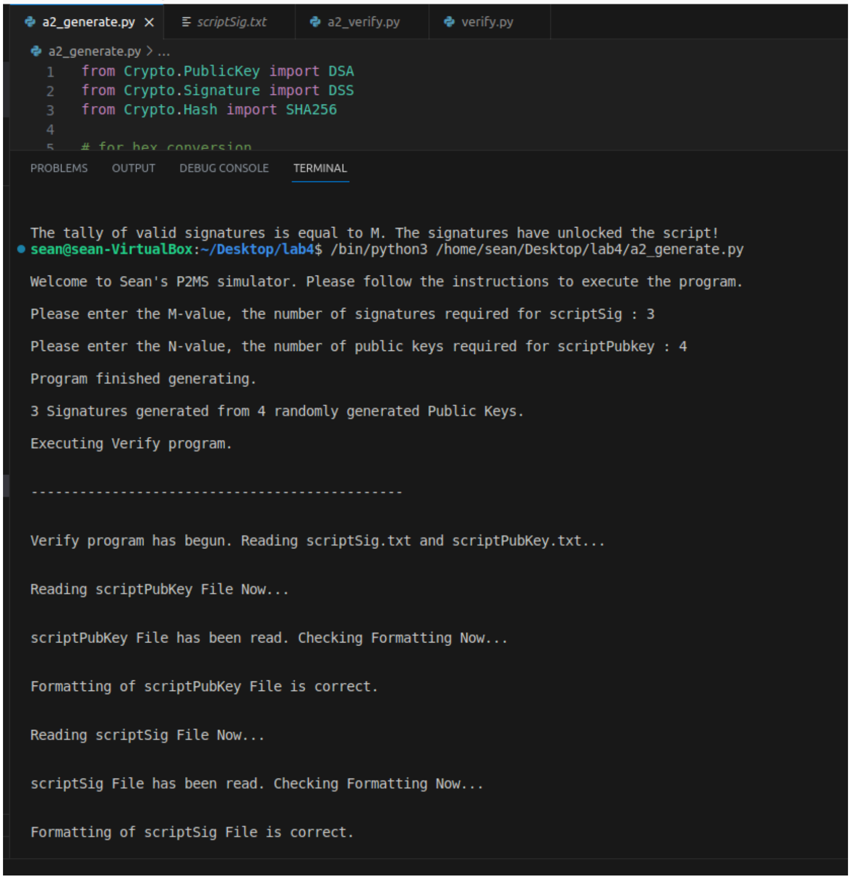
All Necessary information to run the programs:

Additional Python Packages: binascii – for hexadecimal conversion, copy – for deep copying of arrays, subprocess – for issuing a command after a2_generate.py finishes executing, to execute a2_verify.py

Expected Outcomes :

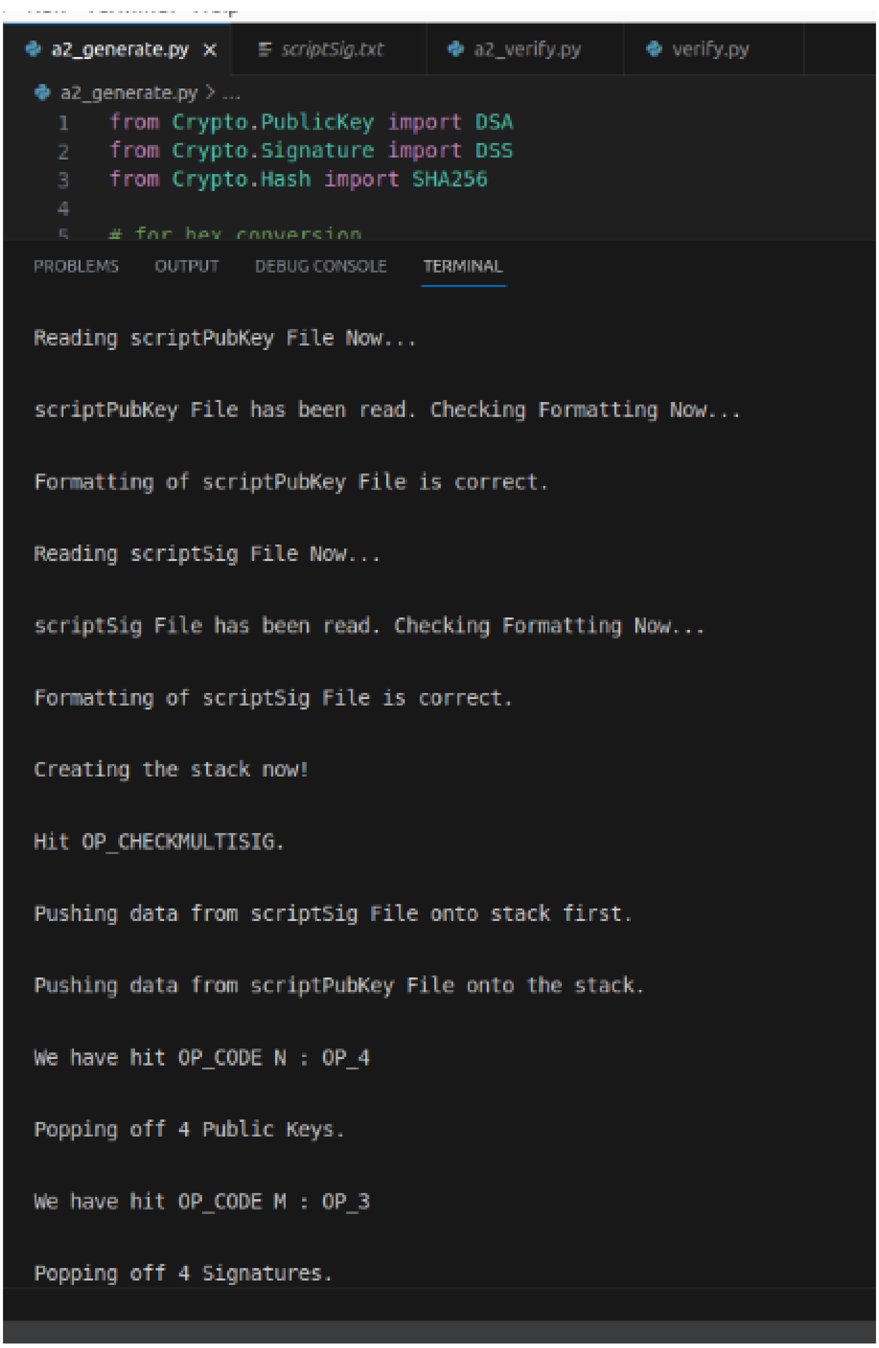
Terminal Output – Only interaction is keying in M and N values

Note: No keys or signatures are shown in the terminal output, only the program logic.



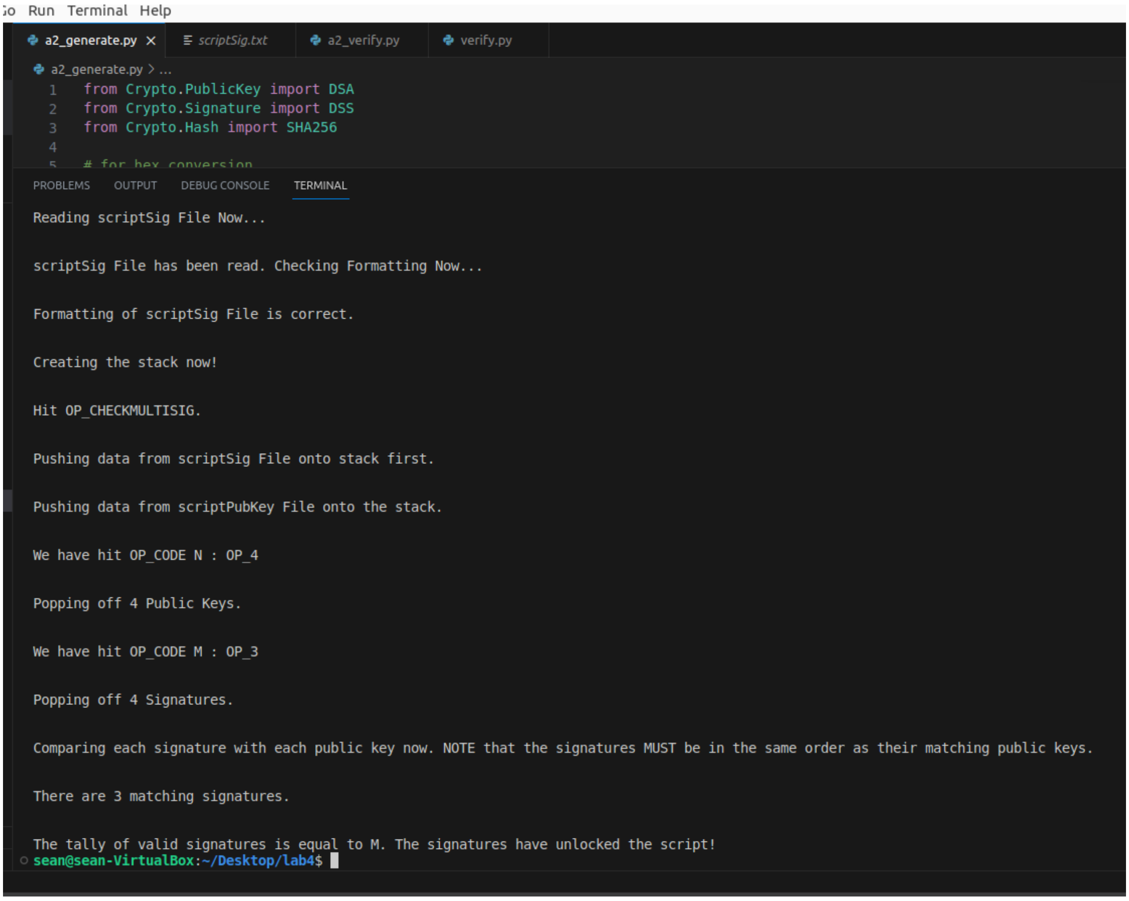
```python
from Crypto.PublicKey import DSA
from Crypto.Signature import DSS
from Crypto.Hash import SHA256

# for hex conversion
```

```
The tally of valid signatures is equal to M. The signatures have unlocked the script!
sean@sean-VirtualBox:~/Desktop/lab4$ /bin/python3 /home/sean/Desktop/lab4/a2_generate.py

Welcome to Sean's P2MS simulator. Please follow the instructions to execute the program.

Please enter the M-value, the number of signatures required for scriptSig : 3

Please enter the N-value, the number of public keys required for scriptPubkey : 4

Program finished generating.

3 Signatures generated from 4 randomly generated Public Keys.

Executing Verify program.


----------------------------------------------


Verify program has begun. Reading scriptSig.txt and scriptPubKey.txt...


Reading scriptPubKey File Now...


scriptPubKey File has been read. Checking Formatting Now...


Formatting of scriptPubKey File is correct.


Reading scriptSig File Now...


scriptSig File has been read. Checking Formatting Now...


Formatting of scriptSig File is correct.
```



```python
from Crypto.PublicKey import DSA
from Crypto.Signature import DSS
from Crypto.Hash import SHA256

# for hex conversion
```

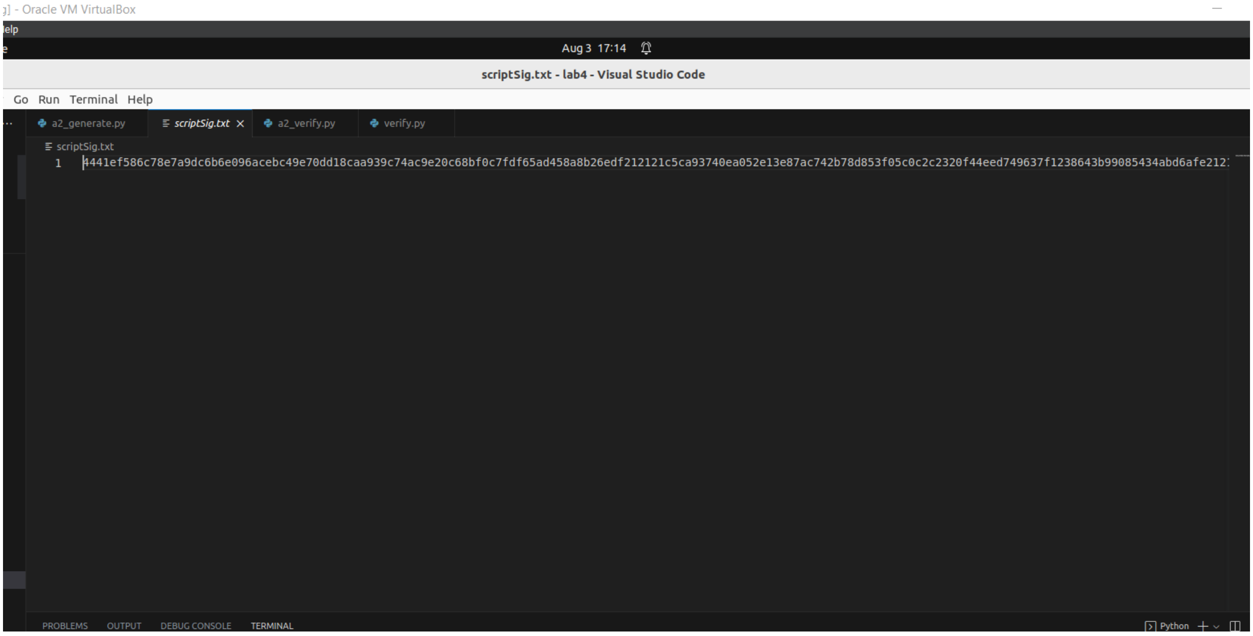
```
Reading scriptPubKey File Now...

scriptPubKey File has been read. Checking Formatting Now...

Formatting of scriptPubKey File is correct.

Reading scriptSig File Now...

scriptSig File has been read. Checking Formatting Now...

Formatting of scriptSig File is correct.

Creating the stack now!

Hit OP_CHECKMULTISIG.

Pushing data from scriptSig File onto stack first.

Pushing data from scriptPubKey File onto the stack.

We have hit OP_CODE N : OP_4

Popping off 4 Public Keys.

We have hit OP_CODE M : OP_3

Popping off 4 Signatures.
```



```
from Crypto.PublicKey import DSA
from Crypto.Signature import DSS
from Crypto.Hash import SHA256

# for hex conversion
```

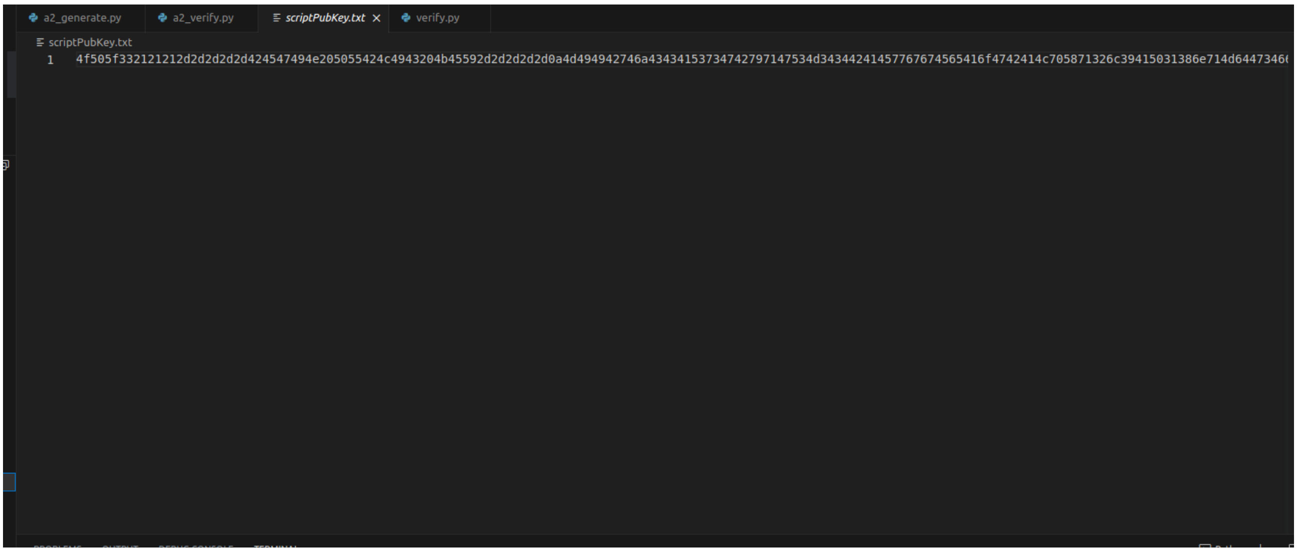
```
Reading scriptSig File Now...

scriptSig File has been read. Checking Formatting Now...

Formatting of scriptSig File is correct.

Creating the stack now!

Hit OP_CHECKMULTISIG.

Pushing data from scriptSig File onto stack first.

Pushing data from scriptPubKey File onto the stack.

We have hit OP_CODE N : OP_4

Popping off 4 Public Keys.

We have hit OP_CODE M : OP_3

Popping off 4 Signatures.

Comparing each signature with each public key now. NOTE that the signatures MUST be in the same order as their matching public keys.

There are 3 matching signatures.

The tally of valid signatures is equal to M. The signatures have unlocked the script!
sean@sean-VirtualBox:~/Desktop/lab4$
```

Output Files – scriptSig.txt



```
4441ef586c78e7a9dc6b6e096acebc49e70dd18caa939c74ac9e20c68bf0c7fdf65ad458a8b26edf212121c5ca93740ea052e13e87ac742b78d853f05c0c2c2320f44eed749637f1238643b99085434abd6afe212
```

Output Files – scriptPubKey.txt



```
4f505f332121212d2d2d2d2d424547494e205055424c4943204b45592d2d2d2d2d0a4d494942746a43434153734742797147534d34344241457767674565416f4742414c705871326c39415031386e714d6447346
```