CS 216: INTRODUCTION TO BLOCKCHAIN

Assignment 2: Bitcoin Scripting

Team Name – CryptoCalypse

Team Members: –

1. Nandan K Prasad (230051012)
2. Srikanth Ravi Jois (230002071)
3. Sai Abhilash Dash (230005041)

1. Introduction: -

This report presents an analysis of Bitcoin scripting through the implementation of transactions using Legacy (P2PKH) and SegWit (P2SH-P2WPKH) address formats. The primary objective of this assignment is to gain hands-on experience with Bitcoin transactions, script validation, and efficiency comparison between the two address types.

The assignment involves creating, signing, and broadcasting transactions on Bitcoin's **regtest** mode using bitcoind and bitcoin-cli through a Python or C-based RPC client.

2. Tools and Environment Setup: -

2.1 Tools Required

- **Bitcoin Core (bitcoind)** – Full node implementation of Bitcoin.
- **Python / C** – Programming language for scripting transactions.
- **Bitcoin CLI (bitcoin-cli)** – Command-line tool to interact with bitcoind.
- **Bitcoin Debugger** – For script validation and debugging.
- **Dependencies:**
  - Python: python-bitcoinlib, bitcoinrpc
  - C: libbitcoin, libcurl

3.Legacy Address Transactions (P2PKH): -

3.1 Steps Implemented

We used the bitcoinrpc library with the rpc_connection functions to generate or load the wallet. Then we generated three legacy addresses linked to the wallet using the same library. The parameter while generating these addresses was ‘legacy’ giving us the standard legacy addresses.

We gave an initial funding to address A before sending 10 BTC through a transaction to the second address B. Then we sent 9.899 BTC to address C.

The transaction id for A to B transaction is
cef2dfc78a33ad54775195e4162ed8c5a76d831ad5c4bea2b32237f3e5903296

While for B to C transaction is
d96797ee98238c80e1e053c760010dcc3916e5886a41cfe45fa2eebe5c3d6bc5

We decoded the transactions to obtain information about the size, the vbytes and the scripts associated with the transactions.

The decoded transacation for A to B is

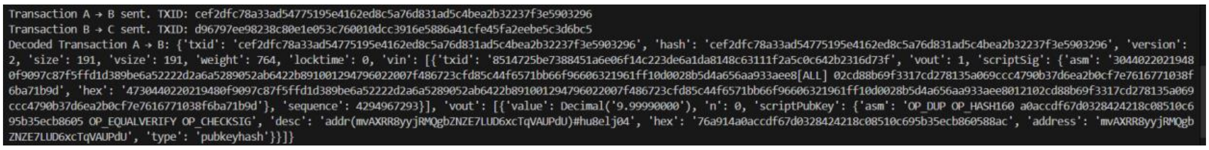
```
Decoded Transaction A → B: {'txid':
'cef2dfc78a33ad54775195e4162ed8c5a76d831ad5c4bea2b32237f3e5903296', 'hash':
'cef2dfc78a33ad54775195e4162ed8c5a76d831ad5c4bea2b32237f3e5903296', 'version': 2,
'size': 191, 'vsize': 191, 'weight': 764, 'locktime': 0, 'vin': [{'txid':
'8514725be7388451a6e06f14c223de6a1da8148c63111f2a5c0c642b2316d73f', 'vout': 1,
'scriptSig': {'asm':
'30440220219480f9097c87f5ffd1d389be6a52222d2a6a5289052ab6422b8910012947960220
07f486723cfd85c44f6571bb66f96606321961ff10d0028b5d4a656aa933aee8[ALL]
02cd88b69f3317cd278135a069ccc4790b37d6ea2b0cf7e7616771038f6ba71b9d', 'hex':
'4730440220219480f9097c87f5ffd1d389be6a52222d2a6a5289052ab6422b89100129479602
2007f486723cfd85c44f6571bb66f96606321961ff10d0028b5d4a656aa933aee8012102cd88b6
9f3317cd278135a069ccc4790b37d6ea2b0cf7e7616771038f6ba71b9d'}, 'sequence':
4294967293}], 'vout': [{'value': Decimal('9.99990000'), 'n': 0, 'scriptPubKey': {'asm':
'OP_DUP OP_HASH160 a0accdf67d0328424218c08510c695b35ecb8605
OP_EQUALVERIFY OP_CHECKSIG', 'desc':
'addr(mvAXRR8yyjRMQgbZNZE7LUD6xcTqVAUPdU)#hu8elj04', 'hex':
'76a914a0accdf67d0328424218c08510c695b35ecb860588ac', 'address':
'mvAXRR8yyjRMQgbZNZE7LUD6xcTqVAUPdU', 'type': 'pubkeyhash'}}]}
```

While for B to C it is

Decoded Transaction B → C: {'txid': 'd96797ee98238c80e1e053c760010dcc3916e5886a41cfe45fa2eebe5c3d6bc5', 'hash': 'd96797ee98238c80e1e053c760010dcc3916e5886a41cfe45fa2eebe5c3d6bc5', 'version': 2, 'size': 191, 'vsize': 191, 'weight': 764, 'locktime': 0, 'vin': [{'txid': 'cef2dfc78a33ad54775195e4162ed8c5a76d831ad5c4bea2b32237f3e5903296', 'vout': 0, 'scriptSig': {'asm': '30440220468834ed3792ee1706f02275f9a20c1fd4478c10e876c8755c0e6266ce9a55f90022008062c60f0fdf6204a951ca4edca4156c43829e77444a7bd4543a1c56e4aeccc[ALL] 0271d827dbee4aa0e760e3bc86682a18948e3b10f536e1319b8993bf9a2f2eddc9', 'hex': '4730440220468834ed3792ee1706f02275f9a20c1fd4478c10e876c8755c0e6266ce9a55f90022008062c60f0fdf6204a951ca4edca4156c43829e77444a7bd4543a1c56e4aeccc01210271d827dbee4aa0e760e3bc86682a18948e3b10f536e1319b8993bf9a2f2eddc9'}, 'sequence': 4294967293}], 'vout': [{'value': Decimal('9.99980000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 15439d4e3e2e9598d0cba7c589ab8a8e4031c26f OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(mhTPV22faoJpossVmYa2j3SetE3znoXzYz)#0rw4eqdn', 'hex': '76a91415439d4e3e2e9598d0cba7c589ab8a8e4031c26f88ac', 'address': 'mhTPV22faoJpossVmYa2j3SetE3znoXzYz', 'type': 'pubkeyhash'}}]}

The challenge script is provided by the vout section of the transaction A to B. It is contained in the asm encoding. This gives the expected hash value the response script must contain to validate the transaction.

The response script gives the hash value upon concatenating [Signature][Public Key][Publick Key Hash] which as given matches with the the challenge script, hence allowing B to unlock its UTXO from the A to B transaction and transact money to address C.



```
Transaction A → B sent. TXID: cef2dfc78a33ad54775195e4162ed8c5a76d831ad5c4bea2b32237f3e5903296
Transaction B → C sent. TXID: d96797ee98238c80e1e053c760010dcc3916e5886a41cfe45fa2eebe5c3d6bc5
Decoded Transaction A → B: {'txid': 'cef2dfc78a33ad54775195e4162ed8c5a76d831ad5c4bea2b32237f3e5903296', 'hash': 'cef2dfc78a33ad54775195e4162ed8c5a76d831ad5c4bea2b32237f3e5903296', 'version':
2, 'size': 191, 'vsize': 191, 'weight': 764, 'locktime': 0, 'vin': [{'txid': '8514725be7388451a6e06f14c223de6a1da8148c63111f2a5c0c642b2316d73f', 'vout': 1, 'scriptSig': {'asm': '3044022021948
0f9097c87f5ffd1d389be6a52222d2a6a5289052ab6422b891001294796022007f486723cfd85c44f6571bb66f96606321961ff10d0028b5d4a656aa933aee8[ALL] 02cd88b69f3317cd278135a069ccc4790b37d6ea2b0cf7e7616771038f
6ba71b9d', 'hex': '4730440220219480f9097c87f5ffd1d389be6a52222d2a6a5289052ab6422b891001294796022007f486723cfd85c44f6571bb66f96606321961ff10d0028b5d4a656aa933aee8012102cd88b69f3317cd278135a069
ccc4790b37d6ea2b0cf7e7616771038f6ba71b9d'}, 'sequence': 4294967293}], 'vout': [{'value': Decimal('9.99990000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 a0accdf67d0328424218c08510c6
95b35ecb8605 OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(mvAXRR8yyjRMQgbZNZE7LUD6xcTqVAUPdU)#hu8elj04', 'hex': '76a914a0accdf67d0328424218c08510c695b35ecb860588ac', 'address': 'mvAXRR8yyjRMQgb
ZNZE7LUD6xcTqVAUPdU', 'type': 'pubkeyhash'}}]}
```

4. SegWit Address Transactions (P2SH): -

4.1 Steps Implemented

The steps are mostly the same as that of legacy address transactions. We still use bitcoinrpc library to generate the addresses. The only difference is that the addresses are used are generated with the ‘p2sh-segwit’ parameter. Hence, they are of the segregated witness variety and not the standard legacy addresses.

After funding address A, we created a transaction to send 10 BTC from address A to address B.

The subsequent transaction id (txid_AtoB) was
3eb780a7abe14a796218af0f3e81cb8799de72ffaf7c69ef5162cc44e0e28e95

This was used as an input in the transaction B to C. This is because we had to verify that the transaction A to B had indeed occurred, meaning that B had unspent transaction outputs from that particular transaction that it could now unlock to send to address C in this new transaction.

This transaction id is
866cf26112a35beb64400d10efbfe42ef14e90326f3b6254d091e95b58e6f077

The decoding of the scripts was done via the same bitcoinrpc library, using rpc_connection functions.

The transaction A to B decoded is

Decoded Transaction A → B: {'txid': '3eb780a7abe14a796218af0f3e81cb8799de72ffaf7c69ef5162cc44e0e28e95', 'hash': '5d2d599d8bc95eb50ca7f0a50913181f9a015f4cc3252368bcc9d056a88d89cf', 'version': 2, 'size': 215, 'vsize': 134, 'weight': 533, 'locktime': 0, 'vin': [{'txid':

```
'fd8ab2ad8f21e919b47bfeb71930554fed478149b5d1d3bc6591c869b53d0cdd', 'vout': 0,
'scriptSig': {'asm': '0014bf21e1e6502dd8ede800cc89ae8faa5b3bdc7215', 'hex':
'160014bf21e1e6502dd8ede800cc89ae8faa5b3bdc7215'}, 'txinwitness':
['304402205b438ae5ad556a09da56dcb16ecc6d940f8525cf0b0bcd2cd9813873bfd232670220
39efa4e313862912e2320f6e069f21934c4902041a24f99392c376fada3f8c9101',
'035761eddc68cce6db97f299148f84a2f05be615443c69a98228f0623ebfa469f9'], 'sequence':
4294967293}], 'vout': [{'value': Decimal('9.99990000'), 'n': 0, 'scriptPubKey': {'asm':
'OP_HASH160 0c0bbfc1521d0dad98ca14f3e534ca1b252416d6 OP_EQUAL', 'desc':
'addr(2MtLvAvXqiHQy2fkoEvEeVX2vLzs751xmCP)#z0kwdxgy', 'hex':
'a9140c0bbfc1521d0dad98ca14f3e534ca1b252416d687', 'address':
'2MtLvAvXqiHQy2fkoEvEeVX2vLzs751xmCP', 'type': 'scripthash'}}]}
```

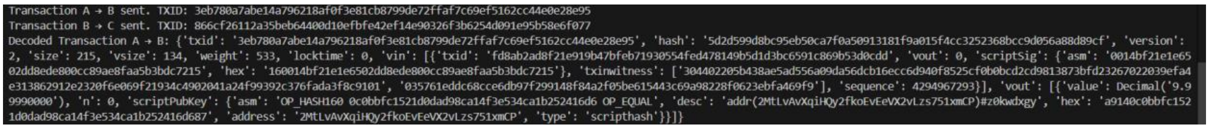
The decoded transaction for B to C is

```
Decoded Transaction B → C: {'txid':
'866cf26112a35beb64400d10efbfe42ef14e90326f3b6254d091e95b58e6f077', 'hash':
'6a4247de0e6e0851d3f1e7fbf3ebf917501af9d74bd7a29c4624100929441865', 'version': 2,
'size': 215, 'vsize': 134, 'weight': 533, 'locktime': 0, 'vin': [{'txid':
'3eb780a7abe14a796218af0f3e81cb8799de72ffaf7c69ef5162cc44e0e28e95', 'vout': 0,
'scriptSig': {'asm': '0014d5082a03937fe757ef054bd1f738d9b0e3d4082b', 'hex':
'160014d5082a03937fe757ef054bd1f738d9b0e3d4082b'}, 'txinwitness':
['304402203c613faba3ac1d517df9536cb47332fec7ebdfac35686917d3b3e52e2a0035cd02204
9e9bd8d5343e87a616eba1e7fa99e3be3d94714f8d0255a7be7468cb21636bd01',
'023df015c72db0ee6ff00815d60ebed85a6163e9663b6ec7e7b060094d06735a10'], 'sequence':
4294967293}], 'vout': [{'value': Decimal('9.99980000'), 'n': 0, 'scriptPubKey': {'asm':
'OP_HASH160 811da225800b1048a22262ecd73b0731ed89850b OP_EQUAL', 'desc':
'addr(2N51vezbCe2wzmDsnAJTxSrZvkidbSQ8QT8)#5n3qw0yz', 'hex':
'a914811da225800b1048a22262ecd73b0731ed89850b87', 'address':
'2N51vezbCe2wzmDsnAJTxSrZvkidbSQ8QT8', 'type': 'scripthash'}}]}
```

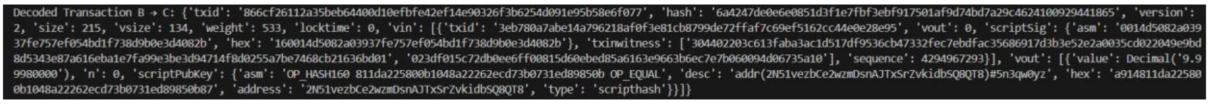
The challenge script is in the vout section of the decoded transaction A to B. The ScriptPubKey encodes the challenge script, possessing the expected hash of the response script after the OP_HASH160.

The response script is in the vin section of the decoded transaction B to C. The concatenation of [Signature][Public Key]{Public Key Hash] provides the hash of the response script which should match the challenge script.

From the above decoded transactions, we can see that the two match (to give 1ba2b3394ff294addc02c6e06af118bc9eae3901)



```
Transaction A → B sent. TXID: 3eb780a7abe14a796218af0f3e81cb8799de72ffaf7c69ef5162cc44e0e28e95
Transaction B → C sent. TXID: 866cf26112a35beb64400d10efbfe42ef14e90326f3b6254d091e95b58e6f077
Decoded Transaction A → B: {'txid': '3eb780a7abe14a796218af0f3e81cb8799de72ffaf7c69ef5162cc44e0e28e95', 'hash': '5d2d599d8bc95eb50ca7f0a50913181f9a015f4cc3252368bcc9d056a88d89cf', 'version':
2, 'size': 215, 'vsize': 134, 'weight': 533, 'locktime': 0, 'vin': [{'txid': 'fd8ab2ad8f21e919b47bfeb71930554fed478149b5d1d3bc6591c869b53d0cdd', 'vout': 0, 'scriptSig': {'asm': '0014bf21e1e65
02dd8ede800cc89ae8faa5b3bdc7215', 'hex': '160014bf21e1e6502dd8ede800cc89ae8faa5b3bdc7215'}, 'txinwitness': ['304402205b438ae5ad556a09da56dcb16ecc6d940f8525cf0b0bcd2cd9813873bfd23267022039efa4
e313862912e2320f6e069f21934c4902041a24f99392c376fada3f8c9101', '035761eddc68cce6db97f299148f84a2f05be615443c69a98228f0623ebfa469f9'], 'sequence': 4294967293}], 'vout': [{'value': Decimal('9.9
9990000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_HASH160 0c0bbfc1521d0dad98ca14f3e534ca1b252416d6 OP_EQUAL', 'desc': 'addr(2MtLvAvXqiHQy2fkoEvEeVX2vLzs751xmCP)#z0kwdxgy', 'hex': 'a9140c0bbfc152
1d0dad98ca14f3e534ca1b252416d687', 'address': '2MtLvAvXqiHQy2fkoEvEeVX2vLzs751xmCP', 'type': 'scripthash'}}]}
```



```
Decoded Transaction B → C: {'txid': '866cf26112a35beb64400d10efbfe42ef14e90326f3b6254d091e95b58e6f077', 'hash': '6a4247de0e6e0851d3f1e7fbf3ebf917501af9d74bd7a29c4624100929441865', 'version':
2, 'size': 215, 'vsize': 134, 'weight': 533, 'locktime': 0, 'vin': [{'txid': '3eb780a7abe14a796218af0f3e81cb8799de72ffaf7c69ef5162cc44e0e28e95', 'vout': 0, 'scriptSig': {'asm': '0014d5082a039
37fe757ef054bd1f738d9b0e3d4082b', 'hex': '160014d5082a03937fe757ef054bd1f738d9b0e3d4082b'}, 'txinwitness': ['304402203c613faba3ac1d517df9536cb47332fec7ebdfac35686917d3b3e52e2a0035cd022049e9bd
8d5343e87a616eba1e7fa99e3be3d94714f8d0255a7be7468cb21636bd01', '023df015c72db0ee6ff00815d60ebed85a6163e9663b6ec7e7b060094d06735a10'], 'sequence': 4294967293}], 'vout': [{'value': Decimal('9.9
9980000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_HASH160 811da225800b1048a22262ecd73b0731ed89850b OP_EQUAL', 'desc': 'addr(2N51vezbCe2wzmDsnAJTxSrZvkidbSQ8QT8)#5n3qw0yz', 'hex': 'a914811da22580
0b1048a22262ecd73b0731ed89850b87', 'address': '2N51vezbCe2wzmDsnAJTxSrZvkidbSQ8QT8', 'type': 'scripthash'}}]}
```

5. Comparison and Analysis: -

5.1 Transaction Size Comparison

Transaction Type	Size (bytes)	Weight (WU)	Virtual Size (vB)
P2PKH (Legacy)	**191**	**764**	**191**
P2SH-P2WPKH (SegWit)	**215**	**533**	**134**

5.2 Script Structure Differences

- **Legacy (P2PKH):** Uses `ScriptSig` for unlocking transactions.
- **SegWit (P2SH-P2WPKH):** Uses witness data, reducing transaction size.

5.3 Benefits of SegWit Transactions

- **Lower Transaction Size:** Reduces transaction fees.
- **Fixes Transaction Malleability:** Prevents modification of transaction hashes.
- **Efficient Block Usage:** Allows more transactions per block.

6. Conclusion

This assignment provides an in-depth exploration of Bitcoin transactions using both **Legacy (P2PKH)** and **SegWit (P2SH-P2WPKH)** formats. By implementing and analyzing transactions, we observed the advantages of SegWit, particularly in terms of reduced transaction size and improved efficiency. The comparative analysis reinforces the importance of SegWit in optimizing Bitcoin's scalability.