BITCOIN SCRIPTING

Github
Team name: sybil servants
Team members:

- Anmol Joshi(230001007)
- Arunav Sameer(230001010)
- Abhitulya Mishra(230002002)

Part 1: Legacy (P2PKH) Address Transactions

Step 1: Connecting to Bitcoin Core RPC and Loading Wallet

- **Action**: The script connects to Bitcoin Core RPC and attempts to load the wallet named `"mywallet"`.
- **RPC Call**: `loadwallet ["mywallet"]`
- **Output**: `{"result":null,"error":{"code":-35,"message":"Wallet \"mywallet\" is already loaded."},"id":1}`
- **Explanation**: The wallet `"mywallet"` is already loaded, indicated by the error code `-35`. In a real scenario, if the wallet wasn't loaded, it would either load it or create it if it didn't exist. Since it's already loaded, the process continues.

Step 2: Generating New Legacy Addresses

- **Action**: Three new legacy (P2PKH) addresses are generated for the wallet.
- **RPC Calls**:
  - `getnewaddress ["", "legacy"]` → Address A: `n3nHxdwYCDf1da11kbu9NPi5ffoW1EH9ga`
  - `getnewaddress ["", "legacy"]` → Address B: `n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b`
  - `getnewaddress ["", "legacy"]` → Address C: `msq1GmHZizz1mruK34vuSg6ipcNwR73jsY`
- **Explanation**: These addresses are created using the legacy format (starting with `m` or `n` in regtest/testnet). They will be used for sending and receiving Bitcoin in subsequent steps.

Step 3: Funding Address A

- **Action**: Send 1.0 BTC to Address A.
- **RPC Call**: `sendtoaddress ["n3nHxdwYCDf1da11kbu9NPi5ffoW1EH9ga", 1.0]`
- **Output**: Transaction ID: `4c6f5ac80f0ebff22ea908b5dd640d40bc2a9c9c1eb087a33b1a81934d860147`
- **Explanation**: This creates a funding transaction that sends 1.0 BTC to Address A. The transaction ID (txid) is returned, which will be used as an input for the next transaction.

Step 4: Generating a Block to Confirm the Funding Transaction

- **Action**: Generate a block to confirm the funding transaction.
- **RPC Calls**:
  - `getnewaddress []` → `bcrt1q8mgsckf8un85fmkk59yzwa6chz35lhzk6n4f68`
  - `generatetoaddress [1, "bcrt1q8mgsckf8un85fmkk59yzwa6chz35lhzk6n4f68"]`
- **Output**: Block hash: `["20a017db35cce9eefdb8127e40c71e51aac263143579f5c7399e987550189744"]`
- **Explanation**: In regtest mode, blocks must be manually generated to confirm transactions. A new address is created, and one block is mined to it, confirming the funding transaction with 1 confirmation.

Step 5: Listing Unspent Outputs (UTXO) for Address A

- **Action**: Check unspent transaction outputs (UTXOs) for Address A.
- **RPC Call**: `listunspent [1, 9999999, ["n3nHxdwYCDf1da11kbu9NPi5ffoW1EH9ga"]]`
- **Output**: A UTXO with txid `4c6f5ac80...`, amount 1.0 BTC, and 1 confirmation.

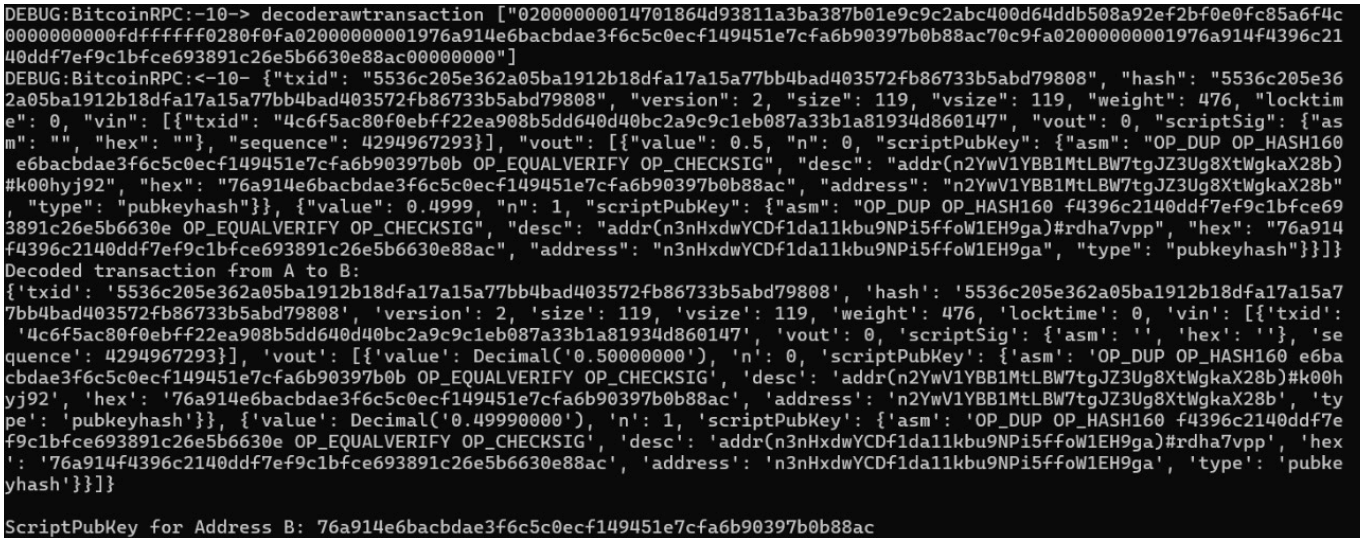
- **Explanation**: The funding transaction created a UTXO of 1.0 BTC at Address A, which is now available to spend. Key details include the txid, vout (0), and scriptPubKey.

Step 6: Creating a Raw Transaction from A to B

- **Action**: Create a raw transaction sending 0.5 BTC to Address B and 0.4999 BTC back to Address A as change.
- **RPC Call**: `createrawtransaction [[{"txid": "4c6f5ac80...", "vout": 0}], {"n2YwV1YBB1...": 0.5, "n3nHxdwYCDf1...": 0.4999}]`
- **Output**: Raw transaction hex: `020000000014701864d...`
- **Explanation**: This constructs an unsigned transaction using the UTXO from Step 5. The total input is 1.0 BTC, with 0.5 BTC sent to Address B and 0.4999 BTC returned to Address A. The difference (0.0001 BTC) is the transaction fee.

Step 7: Decoding the Raw Transaction (A to B)

- **Action**: Decode the raw transaction to verify its structure.
- **RPC Call**: `decoderawtransaction ["020000000014701864d..."]`
- **Output**: Decoded transaction with txid `5536c205e...`, inputs, and outputs (0.5 BTC to Address B, 0.4999 BTC to Address A).
- **Explanation**: The decoded output confirms the transaction structure: one input (from the funding txid) and two outputs. The scriptPubKey for Address B is a P2PKH locking script.



```
DEBUG:BitcoinRPC:-10-> decoderawtransaction ["0200000000014701864d93811a3ba387b01e9c9c2abc400d64ddb508a92ef2bf0e0fc85a6f4c
0000000000fdffffff0280f0fa02000000001976a914e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b88ac70c9fa02000000001976a914f4396c21
40ddf7ef9c1bfce693891c26e5b6630e88ac00000000"]
DEBUG:BitcoinRPC:<-10- {"txid": "5536c205e362a05ba1912b18dfa17a15a77bb4bad403572fb86733b5abd79808", "hash": "5536c205e36
2a05ba1912b18dfa17a15a77bb4bad403572fb86733b5abd79808", "version": 2, "size": 119, "vsize": 119, "weight": 476, "locktim
e": 0, "vin": [{"txid": "4c6f5ac80f0ebff22ea908b5dd640d40bc2a9c9c1eb087a33b1a81934d860147", "vout": 0, "scriptSig": {"as
m": "", "hex": ""}, "sequence": 4294967293}], "vout": [{"value": 0.5, "n": 0, "scriptPubKey": {"asm": "OP_DUP OP_HASH160
 e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b OP_EQUALVERIFY OP_CHECKSIG", "desc": "addr(n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b)
#k00hyj92", "hex": "76a914e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b88ac", "address": "n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b"
, "type": "pubkeyhash"}}, {"value": 0.4999, "n": 1, "scriptPubKey": {"asm": "OP_DUP OP_HASH160 f4396c2140ddf7ef9c1bfce69
3891c26e5b6630e OP_EQUALVERIFY OP_CHECKSIG", "desc": "addr(n3nHxdwYCDf1da11kbu9NPi5ffoW1EH9ga)#rdha7vpp", "hex": "76a914
f4396c2140ddf7ef9c1bfce693891c26e5b6630e88ac", "address": "n3nHxdwYCDf1da11kbu9NPi5ffoW1EH9ga", "type": "pubkeyhash"}}]}
Decoded transaction from A to B:
{'txid': '5536c205e362a05ba1912b18dfa17a15a77bb4bad403572fb86733b5abd79808', 'hash': '5536c205e362a05ba1912b18dfa17a15a7
7bb4bad403572fb86733b5abd79808', 'version': 2, 'size': 119, 'vsize': 119, 'weight': 476, 'locktime': 0, 'vin': [{'txid':
 '4c6f5ac80f0ebff22ea908b5dd640d40bc2a9c9c1eb087a33b1a81934d860147', 'vout': 0, 'scriptSig': {'asm': '', 'hex': ''}, 'se
quence': 4294967293}], 'vout': [{'value': Decimal('0.50000000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 e6ba
cbdae3f6c5c0ecf149451e7cfa6b90397b0b OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b)#k00h
yj92', 'hex': '76a914e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b88ac', 'address': 'n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b', 'ty
pe': 'pubkeyhash'}}, {'value': Decimal('0.49990000'), 'n': 1, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 f4396c2140ddf7e
f9c1bfce693891c26e5b6630e OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(n3nHxdwYCDf1da11kbu9NPi5ffoW1EH9ga)#rdha7vpp', 'hex
': '76a914f4396c2140ddf7ef9c1bfce693891c26e5b6630e88ac', 'address': 'n3nHxdwYCDf1da11kbu9NPi5ffoW1EH9ga', 'type': 'pubke
yhash'}}]}

ScriptPubKey for Address B: 76a914e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b88ac
```

Step 8: Signing the Raw Transaction (A to B)

- **Action**: Sign the raw transaction using the wallet’s private keys.
- **RPC Call**: `signrawtransactionwithwallet ["020000000014701864d..."]`
- **Output**: Signed hex: `020000000014701864d...`, `complete: true`

- **Explanation**: The wallet signs the transaction, adding a scriptSig (signature + public key) to unlock the input. The `complete: true` indicates successful signing.

Step 9: Broadcasting the Transaction (A to B)

- **Action**: Broadcast the signed transaction to the network.
- **RPC Call**: `sendrawtransaction ["0200000000014701864d..."]`
- **Output**: Transaction ID: `7e7e881eb95913bc...`
- **Explanation**: The transaction is successfully broadcast, and a new txid is returned. This txid represents the confirmed transfer from A to B.

Step 10: Generating a Block to Confirm A to B

- **Action**: Generate a block to confirm the A-to-B transaction.
- **RPC Calls**:
  - `getnewaddress []` → `bcrt1quhs39m2...`
  - `generatetoaddress [1, "bcrt1quhs39m2..."]`
- **Output**: Block hash: `["0ba9f22304d91ad1..."]`
- **Explanation**: A new block is mined, confirming the A-to-B transaction with 1 confirmation.

Step 11: Listing UTXO for Address B

- **Action**: Check unspent outputs for Address B.
- **RPC Call**: `listunspent [1, 9999999, ["n2YwV1YBB1..."]]`
- **Output**: UTXO with txid `7e7e881eb...`, amount 0.5 BTC, and 1 confirmation.
- **Explanation**: The A-to-B transaction created a UTXO of 0.5 BTC at Address B, which is now spendable.

Step 12: Creating a Raw Transaction from B to C

- **Action**: Create a raw transaction sending 0.25 BTC to Address C and 0.2499 BTC back to Address B as change.
- **RPC Call**: `createrawtransaction [[{"txid": "7e7e881eb...", "vout": 0}], {"msq1GmHZizz1...": 0.25, "n2YwV1YBB1...": 0.2499}]`
- **Output**: Raw transaction hex: `020000000013974554f...`
- **Explanation**: This constructs an unsigned transaction using the UTXO from Step 11. The total input is 0.5 BTC, with 0.25 BTC sent to Address C and 0.2499 BTC returned to Address B (0.0001 BTC fee).

Step 13: Decoding the Raw Transaction (B to C)

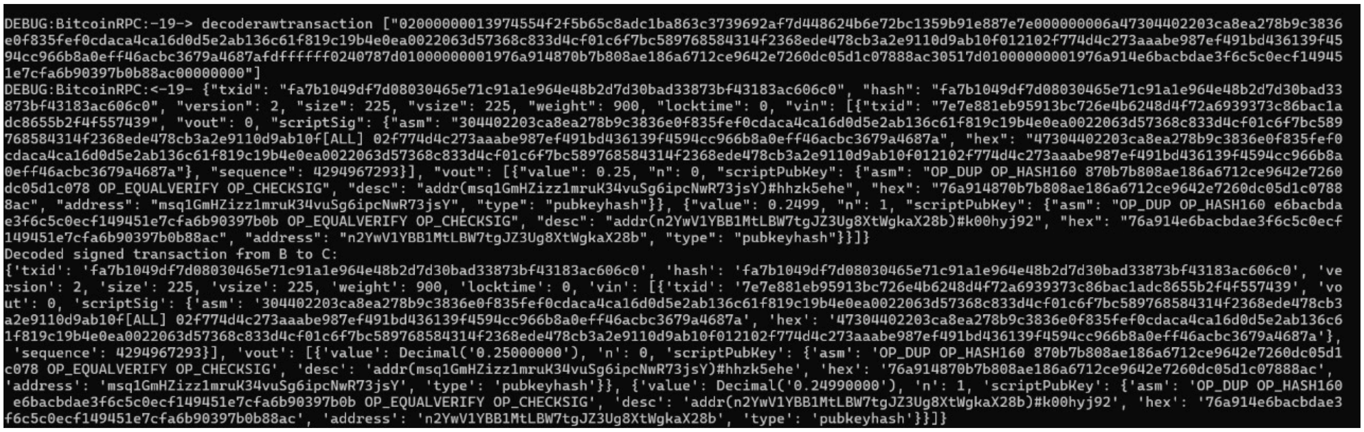
- **Action**: Decode the raw transaction to verify its structure.
- **RPC Call**: `decoderawtransaction ["02000000013974554f..."]`
- **Output**: Decoded transaction with txid `ef39e5ef07...`, inputs, and outputs (0.25 BTC to Address C, 0.2499 BTC to Address B).
- **Explanation**: The decoded output confirms the transaction structure: one input and two outputs.

Step 14: Signing the Raw Transaction (B to C)

- **Action**: Sign the raw transaction using the wallet's private keys.
- **RPC Call**: `signrawtransactionwithwallet ["02000000013974554f..."]`
- **Output**: Signed hex: `02000000013974554f...`, `complete: true`
- **Explanation**: The wallet signs the transaction, adding a scriptSig to unlock the input from Address B.

Step 15: Decoding the Signed Transaction (B to C)

- **Action**: Decode the signed transaction to inspect the scriptSig.
- **RPC Call**: `decoderawtransaction ["02000000013974554f..."]`
- **Output**: Decoded transaction with txid `fa7b1049df...`, including scriptSig.
- **Explanation**: The scriptSig contains a signature and public key, proving ownership of Address B's funds.



```
DEBUG:BitcoinRPC:-19-> decoderawtransaction ["02000000013974554f2f5b65c8adc1ba863c3739692af7d448624b6e72bc1359b91e887e7e000000006a47304402203ca8ea278b9c3836
e0f835fef0cdaca4ca16d0d5e2ab136c61f819c19b4e0ea0022063d57368c833d4cf01c6f7bc589768584314f2368ede478cb3a2e9110d9ab10f012102f774d4c273aaabe987ef491bd436139f45
94cc966b8a0eff46acbc3679a4687afdffffff0240787d01000000001976a914870b7b808ae186a6712ce9642e7260dc05d1c07888ac30517d01000000001976a914e6bacbdae3f6c5c0ecf14945
1e7cfa6b90397b0b88ac00000000"]
DEBUG:BitcoinRPC:<-19- {"txid": "fa7b1049df7d08030465e71c91a1e964e48b2d7d30bad33873bf43183ac606c0", "hash": "fa7b1049df7d08030465e71c91a1e964e48b2d7d30bad33
873bf43183ac606c0", "version": 2, "size": 225, "vsize": 225, "weight": 900, "locktime": 0, "vin": [{"txid": "7e7e881eb95913bc726e4b6248d4f72a6939373c86bac1a
dc8655b2f4f557439", "vout": 0, "scriptSig": {"asm": "304402203ca8ea278b9c3836e0f835fef0cdaca4ca16d0d5e2ab136c61f819c19b4e0ea0022063d57368c833d4cf01c6f7bc589
768584314f2368ede478cb3a2e9110d9ab10f[ALL] 02f774d4c273aaabe987ef491bd436139f4594cc966b8a0eff46acbc3679a4687a", "hex": "47304402203ca8ea278b9c3836e0f835fef0
cdaca4ca16d0d5e2ab136c61f819c19b4e0ea0022063d57368c833d4cf01c6f7bc589768584314f2368ede478cb3a2e9110d9ab10f012102f774d4c273aaabe987ef491bd436139f4594cc966b8a
0eff46acbc3679a4687a"}, "sequence": 4294967293}], "vout": [{"value": 0.25, "n": 0, "scriptPubKey": {"asm": "OP_DUP OP_HASH160 870b7b808ae186a6712ce9642e7260
dc05d1c078 OP_EQUALVERIFY OP_CHECKSIG", "desc": "addr(msq1GmHZizz1mruK34vuSg6ipcNwR73jsY)#hhzk5ehe", "hex": "76a914870b7b808ae186a6712ce9642e7260dc05d1c0788
8ac", "address": "msq1GmHZizz1mruK34vuSg6ipcNwR73jsY", "type": "pubkeyhash"}}, {"value": 0.2499, "n": 1, "scriptPubKey": {"asm": "OP_DUP OP_HASH160 e6bacbda
e3f6c5c0ecf149451e7cfa6b90397b0b OP_EQUALVERIFY OP_CHECKSIG", "desc": "addr(n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b)#k00hyj92", "hex": "76a914e6bacbdae3f6c5c0ecf
149451e7cfa6b90397b0b88ac", "address": "n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b", "type": "pubkeyhash"}}]}
Decoded signed transaction from B to C:
{'txid': 'fa7b1049df7d08030465e71c91a1e964e48b2d7d30bad33873bf43183ac606c0', 'hash': 'fa7b1049df7d08030465e71c91a1e964e48b2d7d30bad33873bf43183ac606c0', 've
rsion': 2, 'size': 225, 'vsize': 225, 'weight': 900, 'locktime': 0, 'vin': [{'txid': '7e7e881eb95913bc726e4b6248d4f72a6939373c86bac1adc8655b2f4f557439', 'vo
ut': 0, 'scriptSig': {'asm': '304402203ca8ea278b9c3836e0f835fef0cdaca4ca16d0d5e2ab136c61f819c19b4e0ea0022063d57368c833d4cf01c6f7bc589768584314f2368ede478cb3
a2e9110d9ab10f[ALL] 02f774d4c273aaabe987ef491bd436139f4594cc966b8a0eff46acbc3679a4687a', 'hex': '47304402203ca8ea278b9c3836e0f835fef0cdaca4ca16d0d5e2ab136c6
1f819c19b4e0ea0022063d57368c833d4cf01c6f7bc589768584314f2368ede478cb3a2e9110d9ab10f012102f774d4c273aaabe987ef491bd436139f4594cc966b8a0eff46acbc3679a4687a'},
 'sequence': 4294967293}], 'vout': [{'value': Decimal('0.25000000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 870b7b808ae186a6712ce9642e7260dc05d1
c078 OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(msq1GmHZizz1mruK34vuSg6ipcNwR73jsY)#hhzk5ehe', 'hex': '76a914870b7b808ae186a6712ce9642e7260dc05d1c07888ac',
'address': 'msq1GmHZizz1mruK34vuSg6ipcNwR73jsY', 'type': 'pubkeyhash'}}, {'value': Decimal('0.24990000'), 'n': 1, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160
 e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b)#k00hyj92', 'hex': '76a914e6bacbdae3
f6c5c0ecf149451e7cfa6b90397b0b88ac', 'address': 'n2YwV1YBB1MtLBW7tgJZ3Ug8XtWgkaX28b', 'type': 'pubkeyhash'}}]}
```

Step 16: Broadcasting the Transaction (B to C)

- **Action**: Broadcast the signed transaction to the network.
- **RPC Call**: `sendrawtransaction ["02000000013974554f..."]`
- **Output**: Transaction ID: `fa7b1049df7d0803...`
- **Explanation**: The transaction is broadcast, transferring funds from B to C.

Step 17: Generating a Block to Confirm B to C

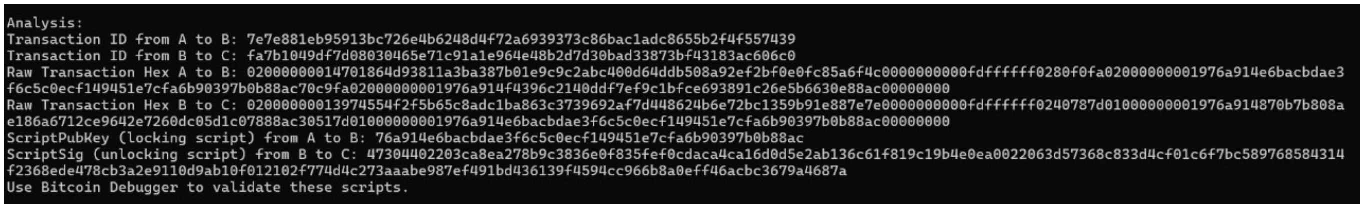
- **Action**: Generate a block to confirm the B-to-C transaction.
- **RPC Calls**:
  - `getnewaddress []` → `bcrt1qz6hh3wp...`
  - `generatetoaddress [1, "bcrt1qz6hh3wp..."]`
- **Output**: Block hash: `["3b19318be438c8f8..."]`
- **Explanation**: A new block confirms the B-to-C transaction.

Step 18: Retrieving the Complete Transaction (B to C)

- **Action**: Fetch the full details of the B-to-C transaction.
- **RPC Call**: `getrawtransaction ["fa7b1049df...", true]`
- **Output**: Complete transaction details, including block hash, confirmations, and timestamps.
- **Explanation**: This confirms the transaction is included in the blockchain with 1 confirmation.

Analysis of Outputs

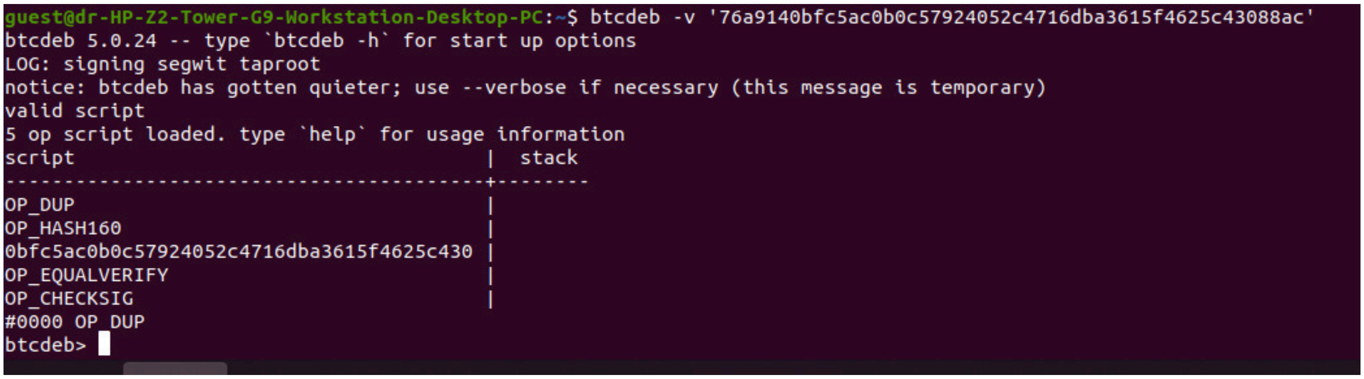
1. **Transaction ID (A to B)**: `7e7e881eb95913bc...`
   - Represents the transfer of 0.5 BTC from Address A to Address B.
2. **Transaction ID (B to C)**: `fa7b1049df7d0803...`
   - Represents the transfer of 0.25 BTC from Address B to Address C.
3. **Raw Transaction Hex (A to B)**: `02000000014701864d...`
   - The unsigned transaction structure before signing.
4. **Raw Transaction Hex (B to C)**: `02000000013974554f...`
   - The unsigned transaction structure before signing.
5. **ScriptPubKey (A to B)**: `76a914e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b88ac`
   - The locking script for Address B, requiring a signature and public key to spend.
6. **ScriptSig (B to C)**: `47304402203ca8ea27...`
   - The unlocking script proving ownership of Address B's funds, consisting of a signature and public key.



```
Analysis:
Transaction ID from A to B: 7e7e881eb95913bc726e4b6248d4f72a6939373c86bac1adc8655b2f4f557439
Transaction ID from B to C: fa7b1049df7d08030465e71c91a1e964e48b2d7d30bad33873bf43183ac606c0
Raw Transaction Hex A to B: 02000000014701864d93811a3ba387b01e9c9c2abc400d64ddb508a92ef2bf0e0fc85a6f4c0000000000fdffffff0280f0fa02000000001976a914e6bacbdae3
f6c5c0ecf149451e7cfa6b90397b0b88ac70c9fa02000000001976a914f4396c2140ddf7ef9c1bfce693891c26e5b6630e88ac00000000
Raw Transaction Hex B to C: 02000000013974554f2f5b65c8adc1ba863c3739692af7d448624b6e72bc1359b91e887e7e0000000000fdffffff0240787d01000000001976a914870b7b808a
e186a6712ce9642e7260dc05d1c07888ac30517d01000000001976a914e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b88ac00000000
ScriptPubKey (locking script) from A to B: 76a914e6bacbdae3f6c5c0ecf149451e7cfa6b90397b0b88ac
ScriptSig (unlocking script) from B to C: 47304402203ca8ea278b9c3836e0f835fef0cdaca4ca16d0d5e2ab136c61f819c19b4e0ea0022063d57368c833d4cf01c6f7bc589768584314
f2368ede478cb3a2e9110d9ab10f012102f774d4c273aaabe987ef491bd436139f4594cc966b8a0eff46acbc3679a4687a
Use Bitcoin Debugger to validate these scripts.
```

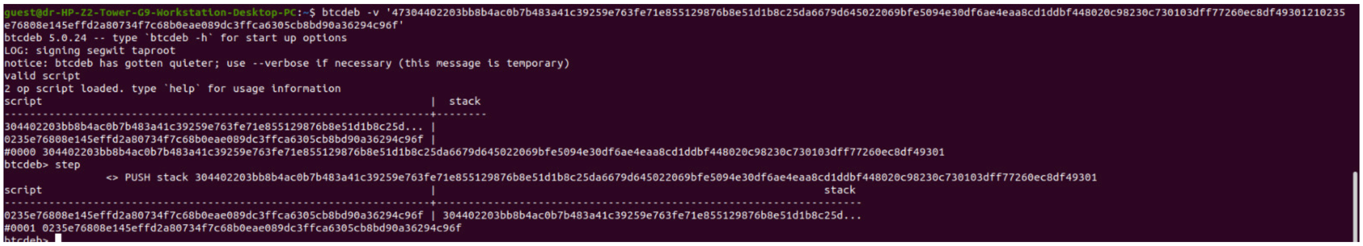
Validation

- The scripts can be validated using a Bitcoin debugger (e.g., `testmempoolaccept` in Bitcoin Core) to ensure the ScriptPubKey and ScriptSig pair correctly. The process involves:
  1. Concatenating the ScriptSig and ScriptPubKey.
  2. Executing the script to verify the signature matches the public key and the public key hashes to the address.



```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb -v '76a9140bfc5ac0b0c57924052c4716dba3615f4625c43088ac'
btcdeb 5.0.24 -- type `btcdeb -h` for start up options
LOG: signing segwit taproot
notice: btcdeb has gotten quieter; use --verbose if necessary (this message is temporary)
valid script
5 op script loaded. type `help` for usage information
script                                   |  stack
-----------------------------------------+--------
OP_DUP                                   |
OP_HASH160                               |
0bfc5ac0b0c57924052c4716dba3615f4625c430 |
OP_EQUALVERIFY                           |
OP_CHECKSIG                              |
#0000 OP_DUP
btcdeb>
```

Validate A to B Locking Script



```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb -v '47304402203bb8b4ac0b7b483a41c39259e763fe71e855129876b8e51d1b8c25da6679d645022069bfe5094e30df6ae4eaa8cd1ddbf448020c98230c730103dff77260ec8df49301210235
e76808e145effd2a80734f7c68b0eae089dc3ffca6305cb8bd90a36294c96f'
btcdeb 5.0.24 -- type `btcdeb -h` for start up options
LOG: signing segwit taproot
notice: btcdeb has gotten quieter; use --verbose if necessary (this message is temporary)
valid script
2 op script loaded. type `help` for usage information
script                                                              |  stack
--------------------------------------------------------------------+--------
304402203bb8b4ac0b7b483a41c39259e763fe71e855129876b8e51d1b8c25d... |
0235e76808e145effd2a80734f7c68b0eae089dc3ffca6305cb8bd90a36294c96f |
#0000 304402203bb8b4ac0b7b483a41c39259e763fe71e855129876b8e51d1b8c25da6679d645022069bfe5094e30df6ae4eaa8cd1ddbf448020c98230c730103dff77260ec8df49301
btcdeb> step
                <> PUSH stack 304402203bb8b4ac0b7b483a41c39259e763fe71e855129876b8e51d1b8c25da6679d645022069bfe5094e30df6ae4eaa8cd1ddbf448020c98230c730103dff77260ec8df49301
script                                                              |                                                             stack
--------------------------------------------------------------------+-------------------------------------------------------------------
0235e76808e145effd2a80734f7c68b0eae089dc3ffca6305cb8bd90a36294c96f | 304402203bb8b4ac0b7b483a41c39259e763fe71e855129876b8e51d1b8c25d...
#0001 0235e76808e145effd2a80734f7c68b0eae089dc3ffca6305cb8bd90a36294c96f
btcdeb>
```

Validate B to C Unlocking Script

Part 2: P2SH- SegWit Address Transactions

Step 1: Connecting to Bitcoin Core RPC and Loading Wallet

- **Action**: The script establishes a connection to Bitcoin Core RPC and attempts to load the wallet named `"mywallet"`.
- **RPC Call**: `loadwallet ["mywallet"]`
- **Output**:

```
{"result": null, "error": {"code": -35, "message": "Wallet \"mywallet\"
is already loaded."}, "id": 1}
```

- **Explanation**: The RPC connection is successful, and the wallet `"mywallet"` is already loaded in the Bitcoin Core instance, as indicated by the error code `-35`. If the wallet weren't loaded, the script would attempt to create it. Since it's already available, the process proceeds.

Step 2: Checking Wallet Balance

- **Action**: Query the wallet's balance to ensure sufficient funds.
- **RPC Call**: `getbalance []`
- **Output**: `649.9990696 BTC`
- **Explanation**: The wallet has a balance of 649.9990696 BTC, which is more than enough to fund the transactions in this scenario (total requirement: 1.0 BTC).

Step 3: Generating P2SH-SegWit Addresses

- **Action**: Generate three new P2SH-SegWit addresses for the wallet.
- **RPC Calls**:
  - `getnewaddress ["", "p2sh-segwit"]` → Address A': `2N8Z8UWYv8bVyxBSKtVEk2ZTrWcohqL8gLG`
  - `getnewaddress ["", "p2sh-segwit"]` → Address B': `2NFHGFMAEreN9CWzKjfXed3yMpDU8M9Sq4c`
  - `getnewaddress ["", "p2sh-segwit"]` → Address C': `2NFf3JpyjSEYgTthE1WwB6si3pWHsVTgcvF`
- **Output**: Three P2SH-SegWit addresses starting with `2`, suitable for regtest/testnet.
- **Explanation**: These addresses use the P2SH-SegWit format (Pay-to-Script-Hash with Segregated Witness), which wraps a SegWit script in a P2SH structure for compatibility with older wallets. They will be used for sending and receiving BTC in the subsequent steps.

Step 4: Funding Address A'

- **Action**: Send 1.0 BTC to Address A' to create a spendable UTXO.
- **RPC Call**: `sendtoaddress ["2N8Z8UWYv8bVyxBSKtVEk2ZTrWcohqL8gLG", 1.0]`
- **Output**: Transaction ID: `f1d7dbca532602f013a74a8b183eccc63158f42637cbe43c644d54e4950fb864`
- **Explanation**: This command creates a funding transaction that transfers 1.0 BTC from the wallet's available funds to Address A'. The returned transaction ID (txid) identifies this

transaction, which will be confirmed in the next step.

Step 5: Generating a Block to Confirm the Funding Transaction

- **Action**: Generate a block to confirm the funding transaction.
- **RPC Calls**:
  - `getnewaddress []` → `bcrt1qgugny8q7ryn6d0fm98aal6vx0yn935ceqhl84q`
  - `generatetoaddress [1, "bcrt1qgugny8q7ryn6d0fm98aal6vx0yn935ceqhl84q"]`
- **Output**: Block hash: `["2232a19faa9a2f631f1ad8edcc8755aa39379af234cc52f0681085db07f09c18"]`
- **Explanation**: In regtest mode, transactions require manual block generation to be confirmed. A new address is created, and one block is mined to it, confirming the funding transaction with 1 confirmation. The block hash identifies the newly mined block.

Step 6: Listing Unspent Outputs (UTXO) for Address A'

- **Action**: Retrieve unspent transaction outputs (UTXOs) for Address A'.
- **RPC Call**: `listunspent [1, 9999999, ["2N8Z8UWYv8bVyxBSKtVEk2ZTrWcohqL8gLG"]]`
- **Output**:

```
[{"txid": "f1d7dbca532602f013a74a8b183ec…", "vout": 1, "address":
"2N8Z8UWYv8bVyxBSKtVEk2ZTrWcohqL8gLG", "amount": 1.0, "confirmations":
1, …}]
```

- **Explanation**: The funding transaction created a UTXO of 1.0 BTC at Address A' (vout 1), confirmed with 1 block. The `redeemScript` is the underlying SegWit script, and the `scriptPubKey` is the P2SH hash of that script. This UTXO is spendable and will be used as the input for the next transaction.

Step 7: Creating a Raw Transaction from A' to B'

- **Action**: Create a raw transaction sending 0.5 BTC to Address B' and 0.4999 BTC back to Address A' as change.
- **RPC Call**:

```
createrawtransaction [[{"txid":
"f1d7dbca532602f013a74a8b183eccc63158f42637cbe43c644d54e4950fb864",
"vout": 1}], {"2NFHGFMAEreN9CWzKjfXed3yMpDU8M9Sq4c": 0.5,
"2N8Z8UWYv8bVyxBSKtVEk2ZTrWcohqL8gLG": 0.4999}]
```

- **Output**: Raw transaction hex: `0200000000164b80f95e4544d643ce4cb3726f45831c6cc3e188b4aa713f0022653cadbd7f10100000000fdffffff0280f0fa020000000017a914f1b57c49726838548d84326b75ffd2e594807b798770c9fa020000000017a914a7ecc5926b8f03a508971422d0ee6cc363950e2d8700000000`
- **Explanation**: This constructs an unsigned transaction using the UTXO from Address A'. The input is 1.0 BTC, with outputs of 0.5 BTC to Address B' and 0.4999 BTC back to Address A'. The difference (0.0001 BTC) is implicitly the transaction fee.

Step 8: Decoding the Raw Transaction (A' to B')

- **Action**: Decode the raw transaction to verify its structure.
- **RPC Call**:

```
decoderawtransaction
["020000000164b80f95e4544d643ce4cb3726f45831c6cc3e188b4aa713f0022653cadb
d7f10100000000fdffffff0280f0fa020000000017a914f1b57c49726838548d84326b75
ffd2e594807b798770c9fa020000000017a914a7ecc5926b8f03a508971422d0ee6cc363
950e2d8700000000"]
```

- **Output**:

```
{"txid":
"cd1d7807294fbeb3a3a255643648abb025f8303ca1940bb26b6255d3d0f0c8b9",
"version": 2, "size": 115, "vsize": 115, "weight": 460, "locktime": 0,
"vin": [{"txid": "f1d7dbca532602f01…", …}], "vout": [{"value": 0.5, …},
{"value": 0.4999, …}]}
```

- **Explanation**: The decoded transaction confirms the structure: one input from the funding txid (vout 1) and two outputs. The `scriptPubKey` for Address B' is a P2SH script (`OP_HASH160 ... OP_EQUAL`), which locks 0.5 BTC to Address B'. The size is 115 bytes (unsigned), and the virtual size (vsize) is also 115 bytes at this stage (before witness data is added).

Step 9: Signing the Raw Transaction (A' to B')

- **Action**: Sign the raw transaction using the wallet's private keys.
- **RPC Call**:

```
signrawtransactionwithwallet
["020000000164b80f95e4544d643ce4cb3726f45831c6cc3e188b4aa713f0022653cadb
d7f10100000000fdffffff0280f0fa020000000017a914f1b57c49726838548d84326b75
ffd2e594807b798770c9fa020000000017a914a7ecc5926b8f03a508971422d0ee6cc363
950e2d8700000000"]
```

- **Output**:

```
{"hex":
"020000000010164b80f95e4544d643ce4cb3726f45831c6cc3e188b4aa713f0022653c
adbd7f1010000000171600142e789a2366048d71b1fa171dfb1b77fcfb87101dfdffffff0
280f0fa020000000017a914f1b57c49726838548d84326b75ffd2e594807b798770c9fa0
20000000017a914a7ecc5926b8f03a508971422d0ee6cc363950e2d870247304402200768059feb8eaf8a1b5c636a35456ba15e36ca1ff0086672cb429e84cbfec9d3022023a196b
24aea2401dc002ea6869e19d4aa2c2448816af5f2c52dcb78b878a1df012102561fc5da3
ed41bc1e28b321b63dd6103d51e9e4dcdf5cd8ce4f43323c907971000000000",
"complete": true}
```

- **Explanation**: The wallet signs the transaction, adding a `scriptSig` (redeem script: `1600142e789a...`) and witness data (signature + public key). The `complete: true` indicates successful signing. The signed transaction now includes SegWit-specific witness data, increasing its size.

Step 10: Broadcasting the Transaction (A' to B')

- **Action**: Broadcast the signed transaction to the network.
- **RPC Call**:

```
sendrawtransaction
["020000000010164b80f95e4544d643ce4cb3726f45831c6cc3e188b4aa713f0022653
cadbd7f1010000000171600142e789a2366048d71b1fa171dfb1b77fcfb87101dfdffffff
0280f0fa020000000017a914f1b57c49726838548d84326b75ffd2e594807b798770c9fa
020000000017a914a7ecc5926b8f03a508971422d0ee6cc363950e2d8702473044022007
68059feb8eaf8a1b5c636a35456ba15e36ca1ff0086672cb429e84cbfec9d3022023a196
b24aea2401dc002ea6869e19d4aa2c2448816af5f2c52dcb78b878a1df012102561fc5da
3ed41bc1e28b321b63dd6103d51e9e4dcdf5cd8ce4f43323c907971000000000"]
```

- **Output**: Transaction ID: `991adadcabaa5a5590cb78ac21267c759eb10b1179aca1c878eedf8ce606875e`
- **Explanation**: The signed transaction is broadcast to the regtest network, and a new txid is returned, representing the confirmed transfer from A' to B'.

Step 11: Generating a Block to Confirm A' to B'

- **Action**: Generate a block to confirm the A'-to-B' transaction.
- **RPC Calls**:
  - `getnewaddress []` → `bcrt1qgnn9gcxqw4gfkunp6zaexmzr22lcyr3f9anu7v`
  - `generatetoaddress [1, "bcrt1qgnn9gcxqw4gfkunp6zaexmzr22lcyr3f9anu7v"]`
- **Output**: Block hash: `["76e380a1bf99cb9273e38889f56665199132baa9e32694c3ea61fa178e402486"]`
- **Explanation**: A new block is mined, confirming the transaction with 1 confirmation.

Step 12: Retrieving Full Transaction Details (A' to B')

- **Action**: Fetch the complete details of the A'-to-B' transaction.
- **RPC Call**:

```
getrawtransaction
["991adadcabaa5a5590cb78ac21267c759eb10b1179aca1c878eedf8ce606875e",
true]
```

- **Output**:

```
{"txid":
"991adadcabaa5a5590cb78ac21267c759eb10b1179aca1c878eedf8ce606875e",
"hash":
"9380c019afb870bba61d6ba4aadcbf6511dfe93ed94ca64cbee59925a4862e4b",
"version": 2, "size": 247, "vsize": 166, "weight": 661, "locktime": 0,
"vin": [{"txid": "f1d7dbca532602f013…", …}], …}
```

- **Explanation**: This provides the full transaction details, including the `scriptSig` (redeem script) and `txinwitness` (signature + public key). The size is 247 bytes (with witness data), and the virtual size (vsize) is 166 bytes, reflecting SegWit's discount on witness data. The transaction is confirmed in the specified block.

Step 13: Listing UTXO for Address B'

- **Action**: Check unspent outputs for Address B'.
- **RPC Call**:

```
listunspent [1, 9999999, ["2NFHGFMAEreN9CWzKjfXed3yMpDU8M9Sq4c"]]
```

- **Output**:

```
[{"txid": "991adadcabaa5a5590cb78…", "vout": 0, "address":
"2NFHGFMAEreN9CWzKjfXed3yMpDU8M9Sq4c", "amount": 0.5, "confirmations":
1, …}]
```

- **Explanation**: The A'-to-B' transaction created a UTXO of 0.5 BTC at Address B' (vout 0), confirmed with 1 block. This UTXO is now available for spending in the next transaction.

Step 14: Creating a Raw Transaction from B' to C'

- **Action**: Create a raw transaction sending 0.25 BTC to Address C' and 0.2499 BTC back to Address B' as change.
- **RPC Call**:

```
createrawtransaction [[{"txid":
"991adadcabaa5a5590cb78ac21267c759eb10b1179aca1c878eedf8ce606875e",
"vout": 0}], {"2NFf3JpyjSEYgTthE1WwB6si3pWHsVTgcvF": 0.25,
"2NFHGFMAEreN9CWzKjfXed3yMpDU8M9Sq4c": 0.2499}]
```

- **Output**: Raw transaction hex: `02000000015e8706e68cdfee78c8a1ac79110bb19e757c2621ac78cb90555aaaabdcda1a990000000000fdffffff0240787d010000000017a914f5d3d5bad03f75587f411b14ecccdb4389e3f7888730517d010000000017a914f1b57c49726838548d84326b75ffd2e594807b798700000000`
- **Explanation**: This constructs an unsigned transaction using the UTXO from Address B'. The input is 0.5 BTC, with outputs of 0.25 BTC to Address C' and 0.2499 BTC back to Address B'. The 0.0001 BTC difference is the fee.

Step 15: Decoding the Raw Transaction (B' to C')

- **Action**: Decode the raw transaction to verify its structure.
- **RPC Call**:

```
decoderawtransaction
["02000000015e8706e68cdfee78c8a1ac79110bb19e757c2621ac78cb90555aaaabdcda
1a990000000000fdffffff0240787d010000000017a914f5d3d5bad03f75587f411b14ec
ccdb4389e3f7888730517d010000000017a914f1b57c49726838548d84326b75ffd2e594
807b798700000000"]
```

- **Output**:

```
{"txid":
"b33fc98c65b873f33e4468683a11a2026bca84ab0a4e5496e0300526f706a1fa",
"version": 2, "size": 115, "vsize": 115, "weight": 460, "locktime": 0,
"vin": [{"txid": "991adadcabaa5a55…", …}], …}
```

- **Explanation**: The decoded transaction confirms one input and two outputs. The `scriptPubKey` for Address C' locks 0.25 BTC with a P2SH script.

Step 16: Signing the Raw Transaction (B' to C')

- **Action**: Sign the raw transaction using the wallet’s private keys.
- **RPC Call**:

```
signrawtransactionwithwallet
["02000000015e8706e68cdfee78c8a1ac79110bb19e757c2621ac78cb90555aaaabdcda
1a990000000000fdffffff0240787d010000000017a914f5d3d5bad03f75587f411b14ec
ccdb4389e3f7888730517d010000000017a914f1b57c49726838548d84326b75ffd2e594
807b798700000000"]
```

- **Output**:

```
{"hex":
"020000000001015e8706e68cdfee78c8a1ac79110bb19e757c2621ac78cb90555aaaabd
cda1a9900000000171600143be49f3b4ea0f3b00dd5db5de552021410690e26fdffffff0
240787d010000000017a914f5d3d5bad03f75587f411b14ecccdb4389e3f7888730517d0
10000000017a914f1b57c49726838548d84326b75ffd2e594807b798702473044022051b
0d67ca03fa30c9907608a8c973c7161571f38573ca0db4e102223f38912ed022015eb108
58aca8a8fad24722285754ab04549381f505be2f087878307182939800121035ed2226ac
0c5de84fcde93963d726939c3b1fa79b528be5064fc030aa229769200000000",
"complete": true}
```

- **Explanation**: The wallet signs the transaction, adding a `scriptSig` (redeem script: `1600143be49f...`) and witness data. The `complete: true` confirms successful signing.

Step 17: Broadcasting the Transaction (B' to C')

- **Action**: Broadcast the signed transaction to the network.
- **RPC Call**:

```
sendrawtransaction
["020000000001015e8706e68cdfee78c8a1ac79110bb19e757c2621ac78cb90555aaaab
dcda1a9900000000171600143be49f3b4ea0f3b00dd5db5de552021410690e26fdffffff
0240787d010000000017a914f5d3d5bad03f75587f411b14ecccdb4389e3f7888730517d
010000000017a914f1b57c49726838548d84326b75ffd2e594807b798702473044022051
b0d67ca03fa30c9907608a8c973c7161571f38573ca0db4e102223f38912ed022015eb10
858aca8a8fad24722285754ab04549381f505be2f087878307182939800121035ed2226a
c0c5de84fcde93963d726939c3b1fa79b528be5064fc030aa229769200000000"]
```

- **Output**: Transaction ID: `aaed8f314d732521934dfaef452f02cc6285bee388cb6afed901563d98fd3611`
- **Explanation**: The transaction is broadcast, transferring funds from B' to C'.

Step 18: Generating a Block to Confirm B' to C'

- **Action**: Generate a block to confirm the B'-to-C' transaction.
- **RPC Calls**:
    - `getnewaddress []` → `bcrt1qj87drg6rmat2mseeervvn8a7qrqejquyp8cpqy`
    - `generatetoaddress [1, "bcrt1qj87drg6rmat2mseeervvn8a7qrqejquyp8cpqy"]`
- **Output**: Block hash: `["5e2a1a1f8d617b04d65e336252775f55c724fc35610601e028355053d81fb5e8"]`
- **Explanation**: A new block confirms the transaction.

Step 19: Retrieving Full Transaction Details (B' to C')

- **Action**: Fetch the complete details of the B'-to-C' transaction.
- **RPC Call**:

```
getrawtransaction
["aaed8f314d732521934dfaef452f02cc6285bee388cb6afed901563d98fd3611",
true]
```

- **Output**:

```
{"txid":
"aaed8f314d732521934dfaef452f02cc6285bee388cb6afed901563d98fd3611",
"hash":
"fa5bac6fbdab2fba5670fb5a10de6c48c99bc08bf40d24721cfaa569c86e73b8",
"version": 2, "size": 247, "vsize": 166, "weight": 661, "locktime": 0,
"vin": [{"txid": "991adadcabaa5a55…", …}], …}
```

- **Explanation**: This confirms the transaction’s inclusion in the blockchain, with size 247 bytes and vsize 166 bytes, reflecting SegWit efficiency.

Step 20: Checking Mempool Info

- **Action**: Query the mempool status.
- **RPC Call**: `getmempoolinfo []`
- **Output**:

  ```
  {"loaded": true, "size": 0, "bytes": 0, "usage": 32, "total_fee": 0.0,
  "maxmempool": 300000000, "mempoolminfee": 1e-05, "minrelaytxfee": 1e-05,
  "incrementalrelayfee": 1e-05, "unbroadcastcount": 0, "fullrbf": false}
  ```

- **Explanation**: The mempool is empty (`size: 0`), as both transactions are confirmed in blocks. The minimum fee is 0.00001 BTC.

Step 21: Checking Blockchain Info

- **Action**: Retrieve blockchain status.
- **RPC Call**: `getblockchaininfo []`
- **Output**:

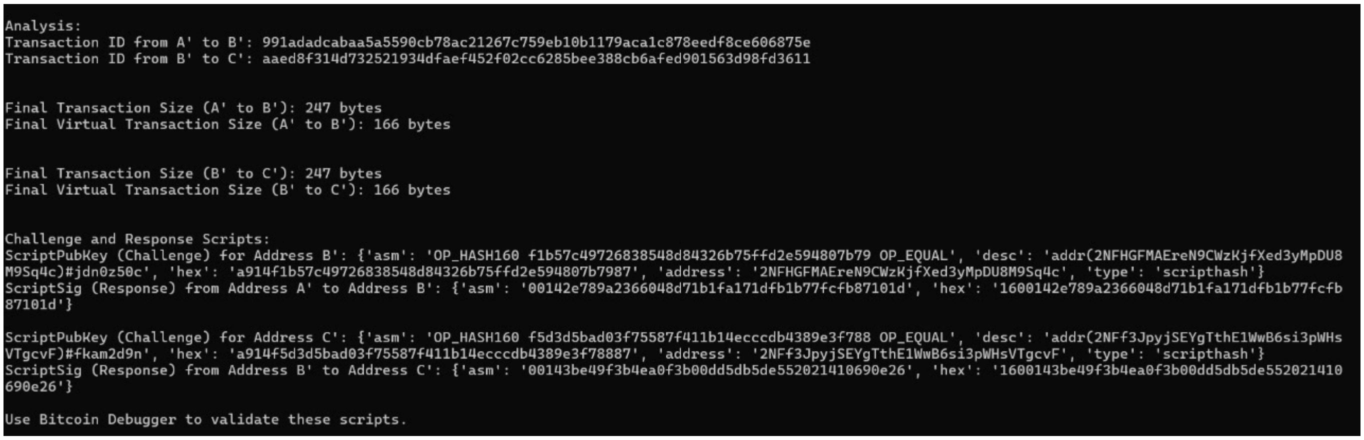
  ```
  {"chain": "regtest", "blocks": 116, "headers": 116, "bestblockhash":
  "5e2a1a1f8d617b04d65e336252775f55c724fc35610601e028355053d81fb5e8",
  "difficulty": 0.0, "time": 1742749967, "mediantime": 1742749678,
  "verificationprogress": 1, "initialblockdownload": false, "chainwork":
  "00000000000000000000000000000000000000000000000000000000000000ea",
  "size_on_disk": 40482, "pruned": false, "warnings": ""}
  ```

- **Explanation**: The regtest blockchain has 116 blocks, with the latest block confirming the B'-to-C' transaction. The difficulty is effectively 0 in regtest mode.

Analysis of Outputs

- **Transaction ID (A' to B')**: `991adadcabaa5a5590cb78ac21267c759eb10b1179aca1c878eedf8ce606875e`
  - Transfers 0.5 BTC from A' to B'.
- **Transaction ID (B' to C')**: `aaed8f314d732521934dfaef452f02cc6285bee388cb6afed901563d98fd3611`
  - Transfers 0.25 BTC from B' to C'.
- **Size and Vsize (A' to B')**: 247 bytes, 166 vbytes

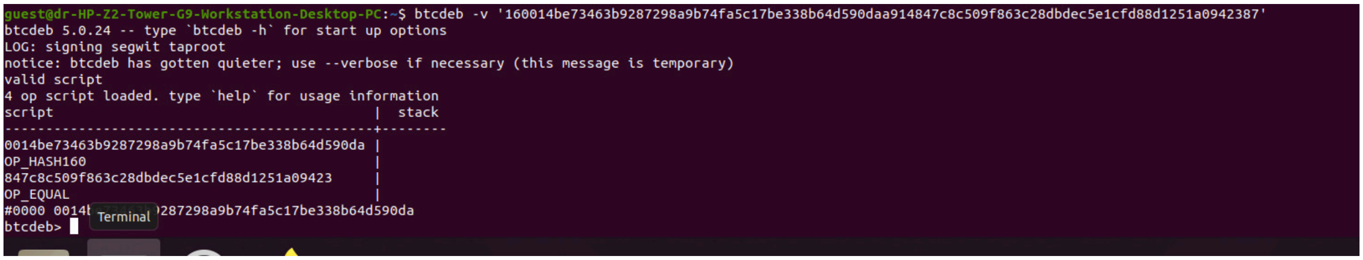
  - Reflects SegWit's separation of witness data, reducing effective size.
- **Size and Vsize (B' to C')**: 247 bytes, 166 vbytes
  - Consistent with A' to B' due to similar structure.
- **Challenge and Response Scripts**:
  - **ScriptPubKey (B')**: `a914f1b57c49726838548d84326b75ffd2e594807b7987` (P2SH locking script)
  - **ScriptSig (A' to B')**: `1600142e789a2366048d71b1fa171dfb1b77fcfb87101d` (redeem script)
  - **ScriptPubKey (C')**: `a914f5d3d5bad03f75587f411b14ecccdb4389e3f78887` (P2SH locking script)
  - **ScriptSig (B' to C')**: `1600143be49f3b4ea0f3b00dd5db5de552021410690e26` (redeem script)



```
Analysis:
Transaction ID from A' to B': 991adadcabaa5a5590cb78ac21267c759eb10b1179aca1c878eedf8ce606875e
Transaction ID from B' to C': aaed8f314d732521934dfaef452f02cc6285bee388cb6afed901563d98fd3611


Final Transaction Size (A' to B'): 247 bytes
Final Virtual Transaction Size (A' to B'): 166 bytes


Final Transaction Size (B' to C'): 247 bytes
Final Virtual Transaction Size (B' to C'): 166 bytes


Challenge and Response Scripts:
ScriptPubKey (Challenge) for Address B': {'asm': 'OP_HASH160 f1b57c49726838548d84326b75ffd2e594807b79 OP_EQUAL', 'desc': 'addr(2NFHGFMAEreN9CWzKjfXed3yMpDU8
M9Sq4c)#jdn0z50c', 'hex': 'a914f1b57c49726838548d84326b75ffd2e594807b7987', 'address': '2NFHGFMAEreN9CWzKjfXed3yMpDU8M9Sq4c', 'type': 'scripthash'}
ScriptSig (Response) from Address A' to Address B': {'asm': '00142e789a2366048d71b1fa171dfb1b77fcfb87101d', 'hex': '1600142e789a2366048d71b1fa171dfb1b77fcfb
87101d'}

ScriptPubKey (Challenge) for Address C': {'asm': 'OP_HASH160 f5d3d5bad03f75587f411b14ecccdb4389e3f788 OP_EQUAL', 'desc': 'addr(2NFf3JpyjSEYgTthE1WwB6si3pWHs
VTgcvF)#fkam2d9n', 'hex': 'a914f5d3d5bad03f75587f411b14ecccdb4389e3f78887', 'address': '2NFf3JpyjSEYgTthE1WwB6si3pWHsVTgcvF', 'type': 'scripthash'}
ScriptSig (Response) from Address B' to Address C': {'asm': '00143be49f3b4ea0f3b00dd5db5de552021410690e26', 'hex': '1600143be49f3b4ea0f3b00dd5db5de552021410
690e26'}

Use Bitcoin Debugger to validate these scripts.
```

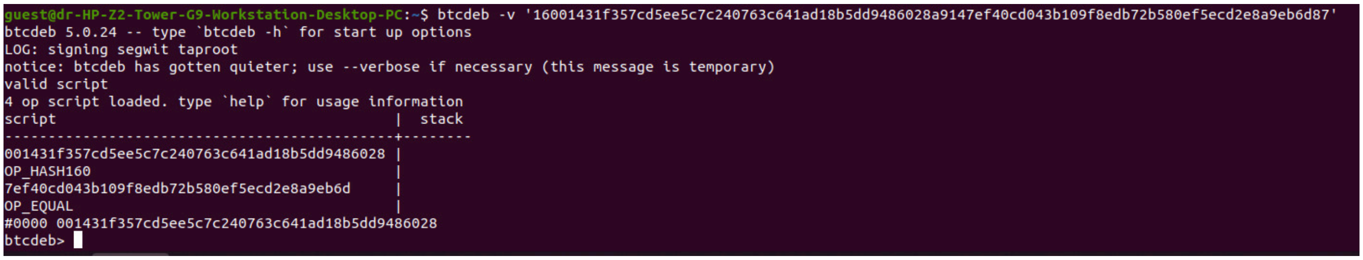
Validation

- The `ScriptPubKey` and `ScriptSig` pairs can be validated using a Bitcoin debugger (e.g., `testmempoolaccept`) by combining them with the witness data to ensure the signature unlocks the funds correctly.



```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb -v '160014be73463b9287298a9b74fa5c17be338b64d590daa914847c8c509f863c28dbdec5e1cfd88d1251a0942387'
btcdeb 5.0.24 -- type `btcdeb -h` for start up options
LOG: signing segwit taproot
notice: btcdeb has gotten quieter; use --verbose if necessary (this message is temporary)
valid script
4 op script loaded. type `help` for usage information
script                                     |  stack
-------------------------------------------+--------
0014be73463b9287298a9b74fa5c17be338b64d590da |
OP_HASH160                                 |
847c8c509f863c28dbdec5e1cfd88d1251a09423     |
OP_EQUAL                                   |
#0000 0014b          287298a9b74fa5c17be338b64d590da
btcdeb> █
```

Validate A to B Locking Script



```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb -v '16001431f357cd5ee5c7c240763c641ad18b5dd9486028a9147ef40cd043b109f8edb72b580ef5ecd2e8a9eb6d87'
btcdeb 5.0.24 -- type `btcdeb -h` for start up options
LOG: signing segwit taproot
notice: btcdeb has gotten quieter; use --verbose if necessary (this message is temporary)
valid script
4 op script loaded. type `help` for usage information
script                                       |  stack
---------------------------------------------+--------
001431f357cd5ee5c7c240763c641ad18b5dd9486028 |
OP_HASH160                                   |
7ef40cd043b109f8edb72b580ef5ecd2e8a9eb6d     |
OP_EQUAL                                     |
#0000 001431f357cd5ee5c7c240763c641ad18b5dd9486028
btcdeb> █
```

Validate B to C Unlocking Script

Part 3: Analysis and Explanation

Metric	P2PKH (Legacy)	P2SH-P2WPKH (SegWit)
Transaction Size (bytes)	225 bytes	247 bytes
Virtual Size (vbytes)	225 vbytes	166 vbytes
Weight (weight units)	900 weight units	661 weight units
ScriptPubKey Size	25 bytes	23 bytes
ScriptSig/Witness Size	106 bytes (ScriptSig)	22 bytes (ScriptSig) + 103 bytes (Witness)

Size comparison

Observation: SegWit transactions (P2SH-P2WPKH) have a larger raw size (247 bytes) due to the inclusion of witness data, but their virtual size (166 vbytes) is significantly smaller than P2PKH transactions (225 vbytes) because witness data is discounted (counted at 1/4 weight). This makes SegWit transactions more efficient in terms of block space usage.

Script Structure Comparison

Legacy Addresses	SegWit Addresses
Signatures and public keys are embedded directly in the transaction’s ScriptSig, bloating the transaction size	Critical validation data (signatures, public keys) is stored in a separate witness field , not counted as heavily toward transaction size
Both the sender and receiver’s public	Only the redeem script hash is

Legacy Addresses	SegWit Addresses
key hashes are stored in the transaction body	embedded in the transaction body, reducing redundancy

Benefit of Segwit Transactions

Why are SegWit transactions smaller?

- Witness Discount: Signature and witness data are counted as only 1/4th the weight of regular transaction data.
- Optimized Scripts: Removes unnecessary opcodes such as OP_DUP and OP_CHECKSIG.
- Separation of Data: Transfers critical information like signatures and public keys to the witness field, reducing the ScriptSig size.

Advantages of SegWit Transactions

- Lower Fees: Smaller transaction size results in reduced transaction costs.
- Scalability: Increased block capacity, allowing more transactions per block.
- Enhanced Security: Prevents transaction malleability by isolating witness data.