

Part 1: Legacy Address Transactions Report

Program Overview

This report details a Python script that uses Bitcoin Core's RPC interface to demonstrate legacy P2PKH transactions. The script performs the following:

1. Connects to a Bitcoin Core node via RPC in regtest mode.
 2. Creates or loads a wallet named "legacywallet".
 3. Generates three legacy P2PKH addresses: A, B, and C.
 4. Executes and broadcasts two transactions: A→B and B→C.
 5. Decodes and analyzes the resulting transaction scripts. The workflow showcases funding, transaction creation, signing, broadcasting, and script validation for legacy P2PKH addresses, which were the standard before SegWit.
-

Workflow and Transaction Details

1. RPC Connection and Wallet Setup

- **RPC Connection:** Established at `http://hashers:xyz111@127.0.0.1:18443` (regtest mode).
- **Wallet:**
 - Name: "legacywallet"
 - Action: Created with `createtimeout` if not present, otherwise loaded with `loadtimeout`.
 - Output: Console confirms wallet creation/loading (e.g., "✓ Wallet 'legacywallet' created successfully").

2. Address Generation

- Generated three legacy P2PKH addresses:
 - A: [addr_A] (e.g., m... on regtest)
 - B: [addr_B]
 - C: [addr_C]
- These addresses use the P2PKH format, starting with "m" or "n" (regtest) or "1" (mainnet).

3. Transaction 1: A → B

- **Funding A:**
 - Mined 101 blocks to A using `generatetoaddress` to mature coinbase outputs.
 - Sent 10 BTC to A via `sendtoaddress`.
 - Funding TXID: [txid_fund].
- **Transaction Details:**

- Amount Sent: 4.9 BTC to B.
- Fee: 0.0001 BTC.
- Change: ~5.0999 BTC returned to A.
- Raw Transaction Hex: [raw_tx].
- Signed Transaction Hex: [signed_tx['hex']].
- Broadcast TXID: [txid_broadcast].
- **Confirmation:** Mined 1 block to C to confirm the transaction.

4. Transaction 2: B → C

- **Input:** UTXO from Transaction 1 (TXID: [txid_broadcast], vout: 0).
 - **Transaction Details:**
 - Amount Sent: 4.8 BTC to C.
 - Fee: 0.0001 BTC.
 - Change: ~0.0999 BTC returned to B.
 - Raw Transaction Hex: [raw_tx_B].
 - Signed Transaction Hex: [signed_tx_B['hex']].
 - Broadcast TXID: [txid_broadcast_B].
 - **Linkage:** The A→B transaction output (UTXO) becomes the input for B→C, chaining the transactions via TXIDs.
-

Decoded Transaction Scripts

Transaction 1 (A → B)

- **Decoded Raw Transaction:**
 - TXID: [decoded_tx_AtoB['txid']]
 - Version: [decoded_tx_AtoB['version']] (e.g., 1)
 - Locktime: [decoded_tx_AtoB['locktime']] (e.g., 0)
 - Outputs: [decoded_tx_AtoB['vout']] (e.g., 4.9 BTC to B, change to A)
- **Locking Script (scriptPubKey) for B: [scriptPubKey_B]**
 - Example: 76a914{20-byte-pubkey-hash}88ac
 -
 -
 -

```

PS C:\Users\komma\OneDrive\bitcoin_ass2> python legacy.py
    ✓ Successfully connected to Bitcoin Core RPC
Chain: regtest, Blocks: 2581
    ⚡ Wallet 'legacy_wallet' is already loaded.

    ✨ Generated Legacy Addresses:
- Address A: mm4qLZuM6Xwo7mwDzyvqdkUkGG8xDEgQEs
- Address B: n2XisobdBdKnpV5VSoAzDqJfy1YuQ2ndrF
- Address C: n3RyDP1hJeerPV8J8oqfEZnetseN8V3Hxc

    💸 Creating transaction from Address A → Address B...

    📄 Raw Transaction (A → B):
0200000001b4cc6a848eedaa5c62d6d44852ca053709b64ec6cfaac7bb0df8b4
eaa3b5e8080000000000fdfffff0280ce341d000000001976a914e67fdb4733
6cd8b3a1b23f8e2c2be6b299ec398d88ac70fc00c01000000160014c47ee64f
dfa0822c7dd500f338fc765091ccc669000000000

    📄 Signed Transaction (A → B):
0200000001b4cc6a848eedaa5c62d6d44852ca053709b64ec6cfaac7bb0df8b4
eaa3b5e808000000006a473044022004a1b4cac033851422e4b4d489f2ffab56
c6a8529e47d4c0c6f41e2efcb8831e02202b486de8ee9c22a452c1f7fb0434eb
e4e8390e1752ac6273007b4a727a1c597e012102e5cf26c0e9fa7ba136d44e46
857b609cd0c8efd8f274c3fbed31525b0b0a9dbfdfffff0280ce341d000000
001976a914e67fdb47336cd8b3a1b23f8e2c2be6b299ec398d88ac70fc00c01
000000160014c47ee64fdfa0822c7dd500f338fc765091ccc669000000000

    ✓ Transaction A → B broadcasted successfully! TX ID: 98789e61c1
5ffed772ff8981b0b5d513799fa5df1f397a2972ba3f5dfb5e79f4

```

-
- **Description:** Decoded output of decoderawtransaction [raw_tx] showing TXID, inputs, outputs, and B's scriptPubKey.

Transaction 2 (B → C)

- **Decoded Raw Transaction:**
 - TXID: [decoded_tx_BtoC['txid']]
 - Version: [decoded_tx_BtoC['version']]
 - Locktime: [decoded_tx_BtoC['locktime']]
 - Outputs: [decoded_tx_BtoC['vout']] (e.g., 4.8 BTC to C, change to B)
- **Locking Script (scriptPubKey) for C:** [scriptPubKey_C]
- **Unlocking Script for B:**
 - scriptSig: [scriptSig_B] (e.g., {signature} {public_key})
- **Description:** Decoded output of decoderawtransaction [raw_tx_B] showing TXID, inputs (referencing [txid_broadcast]), outputs, and scriptSig.

Script Analysis

P2PKH Structure

1. Locking Script (scriptPubKey):

- Format: OP_DUP OP_HASH160 <20-byte pubkey hash> OP_EQUALVERIFY
OP_CHECKSIG
- Hex Example: 76a914{20-byte-pubkey-hash}88ac
- Purpose: Locks funds to a public key hash, requiring a signature from the corresponding private key.

2. Unlocking Script (scriptSig):

- Format: <signature> <public key>
- Example: {72-byte-signature} {33-byte-public-key}
- Purpose: Provides the signature and public key to unlock the UTXO.

3. Validation Mechanism:

- P2PKH: Verifies the public key matches the hash in the scriptPubKey and the signature is valid for the transaction.

Transaction Validation

- A → B: A's wallet signs the input, locking 4.9 BTC to B's P2PKH script.
 - B → C: B unlocks its UTXO with a signature and public key, sending 4.8 BTC to C.
-

Bitcoin Debugger Analysis

Debugger Steps for A → B (Locking Script for B)

- **Input:** scriptPubKey [scriptPubKey_B]
- **Execution:**
 1. OP_DUP: Duplicates the public key.
 2. OP_HASH160: Hashes the public key to a 20-byte hash.
 3. <20-byte-hash>: Pushed to stack.
 4. OP_EQUALVERIFY: Verifies the hash matches.
 5. OP_CHECKSIG: Verifies the signature.
- **Result:** TRUE (valid locking script)

```

guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb '[=020000001b4cc6a848eedaa5c62d6d44852ca053709b64ec6faac7bb0df8b4ea3b5e80800000006a473044022004a1b4cac033851422e4b4d48f2ffab56c6a
8529e47dc0c6f41e2efcb8831e02202b486de8ee9c22a452c1f7fb0434be4e8390e1752ac6273007b4a727a1c597e012102e5c26c0e9fa7ba136d4e46857b69cd0c8ef8f274c3fbed31525b0b0a9db] [76a914e67fdb47336cd83a1
b23f8e2c2be6b299ec398d88ac'
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)
2 op script loaded. type 'help' for usage information
script | stack
-----|-----
3d0323030303030303031623463633613834386565461613563363264366... |
1976a914e67fd47336c08b3a1b2f38e2c2be6b299ec398d88ac'
#0000 3d032303030303031623463633613834386565461613563363264366... |
23230303461316234636330333383513432265346343839663266666162353663361383438656336634161333762623064663862346561613362356383038030303030361343733303434303
3433346562653465383393906531735326163327333030762346137327613163353973653031323130326535636323663053966137626131333643436534363853762363096364306386566643866323734633366265643
3315325623062306139642
btcdeb> stack
- empty stack -
btcdeb step
<> PUSH stack 3d03230303030303162346363361383438656546161356336326436643424383c263613032332738396236465336661616337762623064663862346561613362356383038030303030361343733303434303
038030361343733030303031623463633613834386565461613563363264366434383532636130353337303962364656336634161333762623064663862346561613362356383038030303030361343733303434303
323303034613162346361363033338351343226534634383966326666162353663361383532965343764363863366341613532656632383833165303223032623438364538656539633236134352363166376623
438663237346336662654333155323623062306139642
script | stack
-----|-----
1976a914e67fd47336c08b3a1b2f38e2c2be6b299ec398d88ac |
3d032303030303031623463633613834386565461613563363264366... |
#0000 1976a914e67fd47336c08b3a1b2f38e2c2be6b299ec398d88ac
btcdeb> stack
<> PUSH stack 3d032303030303162346363361383438656546161356336326436643438353263613035333730396236465633663416133377626230646638623465616133623563830380303030361343733303434303
03232303034613162346361363033338351343226534634383966326666162353663361383532965343764363863366341613532656632383833165303223032623438364538656539633236134352363166376623
3033346562653465383393906531735326163327333030762346137327613163353973653031323130326535636323663053966137626131333643436534363853762363096364306386566643866323734633366265643
433135325623062306139642 (top)
btcdeb> |

```

- Description:** Bitcoin debugger output (e.g., btcdeb) showing step-by-step execution of [scriptPubKey_B], ending with TRUE.

Debugger Steps for B → C (Unlocking Script)

```

guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb '[=02000000103a9d05e557438a3aafc51da6cff3896607bfec52dc63ccbd02d3734cbf0b6b00000006a47304402206b5bc148be3a572a950ddca131dbf3ed3d5c6a
880d1b93010793ad34e1d752202207f1f2b47b4ee7625667abe6d559ab49280e8a1451639d0d339b4d0b499c6ea0121034421841c83b168b352461c49153e4c2a8279be772f1c9109378d813aaaa1d400] [76a914f0617476129065d
02000000103a9d05e557438a3aafc51da6cff3896607bfec52dc63ccbd02d3734cbf0b6b00000006a47304402206b5bc148be3a572a950ddca131dbf3ed3d5c68880d1b9301079add3a4e14752202207e1f2b47b4ee7625667abe6
880d2403030303030303031623463633613834386565461613563363264366... |
1976a14f0617476129065d0ec88562ed8d028433a5f86d188ac |
#0000 02000000103a9d05e557438a3aafc51da6cff3896607bfec52dc63ccbd02d3734cbf0b6b00000006a47304402206b5bc148be3a572a950ddca131dbf3ed3d5c68880d1b9301079add3a4e14752202207e1f2b47b4ee7625667abe6
4d559ab49280e8a1451639d0d339b4d0b499c6ea0121034421841c83b168b352461c49153e4c2a8279be772f1c9109378d813aaaa1d400
btcdeb> step
<> PUSH stack 02000000103a9d05e557438a3aafc51da6cff3896607bfec52dc63ccbd02d3734cbf0b6b00000006a47304402206b5bc148be3a572a950ddca131dbf3ed3d5c68880d1b9301079add3a4e14752202207
07e1f2b47b4ee7625667abe6d559ab49280e8a1451639d0d339b4d0b499c6ea0121034421841c83b168b352461c49153e4c2a8279be772f1c9109378d813aaaa1d400
script | stack
-----|-----
1976a14f0617476129065d0ec88562ed8d028433a5f86d188ac |
02000000103a9d05e557438a3aafc51da6cff3896607bfec52dc63ccbd02d...
#0001 02000000103a9d05e557438a3aafc51da6cff3896607bfec52dc63ccbd02d3734cbf0b6b00000006a47304402206b5bc148be3a572a950ddca131dbf3ed3d5c68880d1b9301079add3a4e14752202207e1f2b47b4ee7625667abe6
e64d559ab49280e8a1451639d0d339b4d0b499c6ea0121034421841c83b168b352461c49153e4c2a8279be772f1c9109378d813aaaa1d400 (top)
btcdeb> |

```

- Description:** Debugger output showing stack operations for [scriptSig_B] and [scriptPubKey_B], confirming successful validation.

Conclusion

The script executed two P2PKH transactions:

- A → B:** TXID [txid_broadcast], funded B with 4.9 BTC.
- B → C:** TXID [txid_broadcast_B], spent A→B output to send 4.8 BTC to C. P2PKH provides a simple, widely-used mechanism for Bitcoin transactions. The decoded scripts and debugger steps confirm proper locking (public key hash) and unlocking (signature + public key)

mechanisms. Compared to P2SH-SegWit, P2PKH lacks SegWit's efficiency benefits (e.g., smaller transaction size, malleability fixes).

Part 2: P2SH-SegWit Address Transactions Report

Program Overview

This report details a Python script that uses Bitcoin Core's RPC interface to demonstrate P2SH-SegWit (Pay-to-Script-Hash Segregated Witness) transactions. The script performs the following:

1. Connects to a Bitcoin Core node via RPC
2. Creates or loads a wallet named "testwallet"
3. Generates three P2SH-SegWit addresses: A', B', and C'
4. Executes and broadcasts two transactions: A' → B' and B' → C'
5. Decodes and analyzes the resulting transaction scripts

The workflow showcases funding, transaction creation, signing, broadcasting, and script validation, highlighting the benefits of SegWit and P2SH.

Workflow and Transaction Details

1. RPC Connection and Wallet Setup

- **RPC Connection:** Established at `http://hashers:xyz111@127.0.0.1:18443`
- **Wallet:**
 - Name: "testwallet"
 - Action: Created with `createtimeout` if not present, otherwise loaded with `loadtimeout`
- **Output:** Console confirms wallet creation/loading (e.g., "✓ Wallet 'testwallet' created successfully.")

2. Address Generation

- Generated three P2SH-SegWit addresses:
 - **A':** [addr_Ap] (e.g., 2N... on regtest)
 - **B':** [addr_Bp]
 - **C':** [addr_Cp]
- These addresses use the P2SH-SegWit format, starting with "2" (regtest) or "tb" (testnet).

3. Transaction 1: A' → B'

- **Funding A':**
 - Mined 101 blocks to A' using `generatetoaddress` to mature coinbase outputs
 - Sent 10 BTC to A' via `sendtoaddress`

- Funding TXID: [txid_fund]
- **Transaction Details:**
 - Amount Sent: 4.9 BTC to B'
 - Fee: 0.0001 BTC
 - Change: ~5.0999 BTC returned to A'
 - Raw Transaction Hex: [raw_tx]
 - Signed Transaction Hex: [signed_tx['hex']]
 - Broadcast TXID: [txid_broadcast]
- **Confirmation:** Mined 1 block to C' to confirm the transaction

4. Transaction 2: B' → C'

- **Input:** UTXO from Transaction 1 (TXID: [txid_broadcast], vout: 0)
- **Transaction Details:**
 - Amount Sent: 4.8 BTC to C'
 - Fee: 0.0001 BTC
 - Change: ~0.0999 BTC returned to B'
 - Raw Transaction Hex: [raw_tx_B]
 - Signed Transaction Hex: [signed_tx_B['hex']]
 - Broadcast TXID: [txid_broadcast_B]
- **Linkage:** The A'→B' transaction output becomes the input for B'→C', chaining the transactions via TXIDs.

Decoded Transaction Scripts

Transaction 1 (A' → B')

- **Decoded Raw Transaction:**
 - TXID: [decoded_tx_AtoB['txid']]
 - Version: [decoded_tx_AtoB['version']] (e.g., 1 or 2)
 - Locktime: [decoded_tx_AtoB['locktime']] (e.g., 0)
 - Outputs: [decoded_tx_AtoB['vout']] (e.g., 4.9 BTC to B', change to A')
- **Locking Script (scriptPubKey) for B':** [scriptPubKey_B]
 - Example: a914{20-byte-script-hash}87

```

● Decoded A' → B' Transaction:
- Transaction ID: 8aa12e4e5123aad0b01f42bd986774fbdb2b8b37c936ee35e925af05513140d94
- Version: 2
- Locktime: 0
- Outputs: [{"value": Decimal('4.90000000'), "n": 0, "scriptPubKey": {"asm": "OP_HASH160 df79484cfdf803633346095c5c4b8d7627d4e6c9 OP_EQUAL", "desc": "addr(2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz)#v57n40kn", "hex": "a914df79484cfdf803633346095c5c4b8d7627d4e6c987", "address": "2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz", "type": "scripthash"}, {"value": Decimal('5.09990000'), "n": 1, "scriptPubKey": {"asm": "OP_HASH160 f2f6c2a8dd9072fcbbc72bd1030c95000f452267 OP_EQUAL", "desc": "addr(2NFPu7vGsyztMw8cUBfUnqptoynyhJu8R)#935jjwt", "hex": "a914f2f6c2a8dd9072fcbbc72bd1030c95000f4522678700000000", "address": "2NFPu7vGsyztMw8cUBfUnqptoynyhJu8R", "type": "scripthash"}}]
● komma@pranitha-pc123 MINGW64 ~/OneDrive/bitcoin_ass2
$ bitcoin-cli -regtest decoderawtransaction 0200000001fe4ae6df95f4edef342e37b44e661c2b0d82cf27b85a479585c8d7a55412f340000000000fdfffff0280ce341d00000000017a914df79484cfdf803633346095c5c4b8d7627d4e6c9870d4651e0000000017a914f2f6c2a8dd9072fcbbc72bd1030c95000f4522678700000000
{
  "txid": "8aa12e4e5123aad0b01f42bd986774fbdb2b8b37c936ee35e925af05513140d94",
  "hash": "8aa12e4e5123aad0b01f42bd986774fbdb2b8b37c936ee35e925af05513140d94",
  "version": 2,
  "size": 115,
  "vsize": 115,
  "weight": 460,
  "locktime": 0,
  "vin": [
    {
      "txid": "342f41557a8d5c5879a4857bf22cd8b0c261e6447be342f3eaedf495df64afe",
      "vout": 0,
      "scriptSig": {
        "asm": "",
        "hex": ""
      },
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": 4.90000000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_HASH160 df79484cfdf803633346095c5c4b8d7627d4e6c9 OP_EQUAL",
        "desc": "addr(2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz)#v57n40kn",
        "hex": "a914df79484cfdf803633346095c5c4b8d7627d4e6c987",
        "address": "2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz",
        "type": "scripthash"
      },
      "n": 1,
      "value": 5.09990000,
      "scriptPubKey": {
        "asm": "OP_HASH160 f2f6c2a8dd9072fcbbc72bd1030c95000f452267 OP_EQUAL",
        "address": "2NFPu7vGsyztMw8cUBfUnqptoynyhJu8R",
        "type": "scripthash"
      }
    }
  ]
}
●

```

- Description:** Decoded output of decoderawtransaction [raw_tx] showing TXID, inputs, outputs, and B's scriptPubKey.

Transaction 2 (B' → C')

- Decoded Raw Transaction:**
 - TXID: [decoded_tx_BtoC['txid']]
 - Version: [decoded_tx_BtoC['version']]
 - Locktime: [decoded_tx_BtoC['locktime']]
 - Outputs: [decoded_tx_BtoC['vout']] (e.g., 4.8 BTC to C, change to B')
- Locking Script (scriptPubKey) for C': [scriptPubKey_C]**
- Unlocking Script for B':**
 - scriptSig: [scriptSig_B] (e.g., 160014{20-byte-pubkey-hash})
 - Witness: [scriptWitness_B] (e.g., [signature, public_key])

```

● Decoded B' → C' Transaction:
- Transaction ID: a9704ec981e272be3d144b7c9cece3ba83ee327a34d31bf19edcee5594801783
- Version: 2
- Locktime: 0
- Outputs: [{"value": Decimal('4.80000000'), "n": 0, "scriptPubKey": {"asm": "OP_HASH160 9dd7a287fa4135379237dac131d6efbd1f34bcfc OP_EQUAL", "desc": "addr(2N7dpPyZPstf4HAojJxHlyPwRBzGTZQyIZY)#sa46a4g", "hex": "a9149dd7a287fa4135379237dac131d6efbd1f34bcfc87", "address": "2N7dpPyZPstf4HAojJxHlyPwRBzGTZQyIZY", "type": "scripthash"}, {"value": Decimal('0.09990000'), "n": 1, "scriptPubKey": {"asm": "OP_HASH160 df79484cfdf80363346095c5c4b8d7627d4e6c9 OP_EQUAL", "desc": "addr(2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz)#v57n40kn", "hex": "a914df79484cfdf80363346095c5c4b8d7627d4e6c987", "address": "2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz", "type": "scripthash"}}]
● komma@pranitha-pc123 MINGW64 ~/OneDrive/bitcoin_ass2
$ bitcoin-cli -regtest decoderawtransaction 02000000001b45d76c837632cdce3271c825ea2baa581ecc7beffce25004e8d92799d6987ad0000000000fdfffff0200389c1c0000000
{
  "txid": "a9704ec981e272be3d144b7c9cece3ba83ee327a34d31bf19edcee5594801783",
  "hash": "a9704ec981e272be3d144b7c9cece3ba83ee327a34d31bf19edcee5594801783",
  "version": 2,
  "size": 115,
  "vsize": 115,
  "weight": 460,
  "locktime": 0,
  "vin": [
    {
      "txid": "ad87699d79928d4e0025ceffbec7ec81a5baa25e821c27e3dc2c2c6337c8765db4",
      "vout": 0,
      "scriptSig": {
        "asm": "",
        "hex": ""
      },
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": 4.80000000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_HASH160 9dd7a287fa4135379237dac131d6efbd1f34bcfc OP_EQUAL",
        "desc": "addr(2N7dpPyZPstf4HAojJxHlyPwRBzGTZQyIZY)#sa46a4g",
        "hex": "a9149dd7a287fa4135379237dac131d6efbd1f34bcfc87",
        "address": "2N7dpPyZPstf4HAojJxHlyPwRBzGTZQyIZY",
        "type": "scripthash"
      },
      "n": 1,
      "scriptPubKey": {
        "asm": "OP_HASH160 df79484cfdf80363346095c5c4b8d7627d4e6c9 OP_EQUAL",
        "desc": "addr(2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz)#v57n40kn",
        "hex": "a914df79484cfdf80363346095c5c4b8d7627d4e6c987",
        "address": "2NDcqxutiy3ALT8mWFYsNcnFqjUyG5sr2tz",
        "type": "scripthash"
      }
    }
  ]
}

```

- **Description:** Decoded output of decoderawtransaction [raw_tx_B] showing TXID, inputs (referencing [txid_broadcast]), outputs, scriptSig, and witness data.

Script Analysis

P2SH-SegWit Structure

1. **Locking Script (scriptPubKey):**
 - Format: OP_HASH160 <20-byte script hash> OP_EQUAL
 - Hex Example: a914{script-hash}87
 - Purpose: Locks funds to a script hash, requiring the redeem script to match.
2. **Unlocking Script:**
 - **scriptSig:** Pushes the redeem script (e.g., 0014{public-key-hash} for P2WPKH)
 - **Witness:** Contains signature and public key, separated for SegWit efficiency
 - Example:
 - scriptSig: 16 00 14 {20-byte-hash}
 - Witness: [72-byte-signature, 33-byte-public-key]

3. Validation Mechanism:

- **P2SH**: Verifies the redeem script's hash matches the scriptPubKey.
- **SegWit**: Validates the witness data against the redeem script (e.g., signature matches public key).
- **Execution**: Combines hash verification and signature checking.

Transaction Validation

- **A' → B'**: A's wallet signs the input, locking 4.9 BTC to B's P2SH-SegWit script.
- **B' → C'**: B' unlocks its UTXO with redeem script and witness, sending 4.8 BTC to C'.

Bitcoin Debugger Analysis

Debugger Steps for A' → B' (Locking Script for B')

- **Input**: scriptPubKey [scriptPubKey_B]
- **Execution**:
 - Step 1: OP_HASH160 computes hash of redeem script
 - Step 2: <20-byte-hash> pushed to stack
 - Step 3: OP_EQUAL verifies match
- **Result**: TRUE (valid locking script)

[Insert Screenshot #3 Here]

- *Description*: Bitcoin debugger output (e.g., btcdeb or bitcoin-tx) showing step-by-step execution of [scriptPubKey_B], ending with TRUE.

Debugger Steps for B' → C' (Unlocking Script)

- **Input**: scriptSig [scriptSig_B] + Witness [scriptWitness_B]
- **Execution**:
 - Step 1: scriptSig pushes 0014{public-key-hash}
 - Step 2: Hash of redeem script verified against previous scriptPubKey
 - Step 3: Witness signature validated against public key
- **Result**: TRUE (valid unlocking)

[Insert Screenshot #4 Here]

- *Description*: Debugger output showing stack operations for [scriptSig_B] and [scriptWitness_B], confirming successful validation.

Conclusion

The script executed two P2SH-SegWit transactions:

- **A' → B'**: TXID [txid_broadcast], funded B' with 4.9 BTC

- **B' → C':** TXID [txid_broadcast_B], spent A'→B' output to send 4.8 BTC to C' P2SH-SegWit combines script flexibility with SegWit's efficiency (reduced size, malleability protection). The decoded scripts and debugger steps confirm proper locking and unlocking mechanisms.