CS- 216 INTRODUCTION TO BLOCKCHAIN

BITCOIN SCRIPTING

Team Members:

- PARAM SAXENA (230001060)
- JAGRIT (230051005)
- SAUMYA VAIDYA (230008035)

INTRODUCTION

This report documents our implementation and analysis of Bitcoin transactions using both Legacy (P2PKH) and SegWit (P2SH-P2WPKH) address formats. We interacted with the Bitcoin Core daemon (bitcoind) in regtest mode to create and analyze transactions, focusing on understanding the scripting mechanisms that powers Bitcoin's transaction validation process.

Environment Setup

We used the following environment for this assignment:

- Bitcoin Core v25.0 in regtest mode
- Python 3.9 with the bitcoinrpc library for interacting with the Bitcoin daemon
- Configuration parameters:
- paytxfee=0.0001fallbackfee=0.0002mintxfee=0.00001txconfirmtarget=1

Part 1: Legacy P2PKH Transactions

WORKFLOW

Transaction from A to B

- **TXID**:1debec1503fd9923aaa2eda825d9062a7d439b285a20d6aa63ae18ab393e3b96
- This transaction represents a transfer from Address A to Address B. The output (UTXO) from this transaction is what Address B spends in the next transaction. From the provided data, we see this UTXO listed with an amount of 1 BTC, confirmed once, and tied to Address B (mmZLETCXugXivJk4B8ZMCEoX9FUFwHhfQz). The scriptPubKey (locking script) for this UTXO is: 76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac
- This UTXO becomes the input for the subsequent transaction from B to C.

Transaction from B to C

- **TXID**: d327950aeff12dd6f40b186318e317c8c43eb778b6f2471096c67ae5bfb99021
- This transaction spends the UTXO from the A-to-B transaction. It takes the 1 BTC input and splits it into two outputs:
    - 0.5 BTC to Address C (mhz17c1z5RQU3k3dn4TBixyEhpoo7oV2qQ)
    - 0.4999 BTC back to Address B (change).
- The input references the previous transaction's TXID (1debec1503fd9923...) and output index (vout: 0), unlocking it with a scriptSig.

DECODED SCRIPTS

Transaction A to B (Input for B to C)

- **ScriptPubKey (Challenge Script)**
    - Hex: 76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac
    - ASM: OP_DUP OP_HASH160 4243e6437c5cbee99814cea6f22f6c6b52b0cefa OP_EQUALVERIFY OP_CHECKSIG

- This is a standard Pay-to-PubKeyHash (P2PKH) locking script. It locks the 1 BTC to Address B, requiring a signature and public key to unlock it.

(BELOW IS THE SCREENSHOT OF DECODED TX A to B)

```
Decoded transaction:
{
  "txid": "1debec1503fd9923aaa2eda825d9062a7d439b285a20d6aa63ae18ab393e3b96",
  "hash": "1debec1503fd9923aaa2eda825d9062a7d439b285a20d6aa63ae18ab393e3b96",
  "version": 2,
  "size": 225,
  "vsize": 225,
  "weight": 900,
  "locktime": 0,
  "vin": [
    {
      "txid": "35f00d379598ba8e08b0155e4bc858eb33d35506c7754009fa51193e0010bad2",
      "vout": 0,
      "scriptSig": {
        "asm": "3044022016d51234fab89591b5f5aa7cdb09855148914 9e415b71d52c96ef8da18905f97022065493dca13ced8fd6f4074f2bb1a9b1e55e836e2aa63ff1413111e918899ee9e[ALL] 03df02386aaa7
        71daff35d3e8c4a96f6b1297a2e8fc35f55c622b49c689e31c616",
        "hex": "473044022016d51234fab89591b5f5aa7cdb09855148914 9e415b71d52c96ef8da18905f97022065493dca13ced8fd6f4074f2bb1a9b1e55e836e2aa63ff1413111e918899ee9e012103df02386aaa7
        71daff35d3e8c4a96f6b1297a2e8fc35f55c622b49c689e31c616"
      },
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": "1.00000000",
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_DUP OP_HASH160 4243e6437c5cbee99814cea6f22f6c6b52b0cefa OP_EQUALVERIFY OP_CHECKSIG",
        "desc": "addr(mmZLETCXugXivJk4B8ZMCEoX9FUFwHhfQz)#frft054q",
        "hex": "76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac",
        "address": "mmZLETCXugXivJk4B8ZMCEoX9FUFwHhfQz",
        "type": "pubkeyhash"
      }
    },
    {
      "value": "24.00000000",
      "n": 1,
      "scriptPubKey": {
        "asm": "OP_DUP OP_HASH160 d3dd2b69030ae0426ed6eba1710f06d456d4332b OP_EQUALVERIFY OP_CHECKSIG",
        "desc": "addr(mzqBpfKMHNNMTsqhctEozjHpKUfWcgPd1L)#6qxzmytk",
        "hex": "76a914d3dd2b69030ae0426ed6eba1710f06d456d4332b88ac",
        "address": "mzqBpfKMHNNMTsqhctEozjHpKUfWcgPd1L",
        "type": "pubkeyhash"
      }
    }
  ]
```


Transaction B to C

- **ScriptSig (Response Script)**
  - ASM:304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655acb03d38a2d58e[ALL] 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
  - Hex:47304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655acb03d38a2d58e012103cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86

- Breakdown:
  - 30440220...d58e: A 71-byte DER-encoded signature (with SIGHASH_ALL).
  - 03cc8e58...bb86: A 33-byte compressed public key.

(BELOW IS THE SCREENSHOT OF DECODED TX B to C)

```
Decoded transaction:
{
  "txid": "d327950aeff12dd6f40b186318e317c8c43eb778b6f2471096c67ae5bfb99021",
  "hash": "d327950aeff12dd6f40b186318e317c8c43eb778b6f2471096c67ae5bfb99021",
  "version": 2,
  "size": 225,
  "vsize": 225,
  "weight": 900,
  "locktime": 0,
  "vin": [
    {
      "txid": "1debec1503fd9923aaa2eda825d9062a7d439b285a20d6aa63ae18ab393e3b96",
      "vout": 0,
      "scriptSig": {
        "asm": "304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655acb03d38a2d58e[ALL] 03cc8e58e4a
        e5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86",
        "hex": "47304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655acb03d38a2d58e012103cc8e58e4a
        e5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86"
      },
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": 0.5,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_DUP OP_HASH160 1b0dd61f6e9a1929cff7c91333afc03e17f308da OP_EQUALVERIFY OP_CHECKSIG",
        "desc": "addr(mhz17c1z5RQU3k3dn4TBixyEhpoo7oV2qQ)#ptxmcr3z",
        "hex": "76a9141b0dd61f6e9a1929cff7c91333afc03e17f308da88ac",
        "address": "mhz17c1z5RQU3k3dn4TBixyEhpoo7oV2qQ",
        "type": "pubkeyhash"
      }
    },
    {
      "value": 0.4999,
      "n": 1,
      "scriptPubKey": {
        "asm": "OP_DUP OP_HASH160 4243e6437c5cbee99814cea6f22f6c6b52b0cefa OP_EQUALVERIFY OP_CHECKSIG",
        "desc": "addr(mmZLETCXugXivJk4B8ZMCEoX9FUFwHhfQz)#frft054q",
        "hex": "76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac",
        "address": "mmZLETCXugXivJk4B8ZMCEoX9FUFwHhfQz",
        "type": "pubkeyhash"
      }
    }
  ]
}
```


- **ScriptPubKey (Outputs)**
  - Output 0 (to Address C):
    - Hex: 76a9141b0dd61f6e9a1929cff7c91333afc03e17f308da88ac
    - ASM: OP_DUP OP_HASH160 1b0dd61f6e9a1929cff7c91333afc03e17f308da OP_EQUALVERIFY OP_CHECKSIG
  - Output 1 (change to Address B):
    - Hex: 76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac

        - ASM: OP_DUP OP_HASH160 4243e6437c5cbee99814cea6f22f6c6b52b0cefa OP_EQUALVERIFY OP_CHECKSIG

SCRIPT STRUCTURE AND VALIDATION MECHANISM

Challenge Script (scriptPubKey from A to B)

- **Structure**: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
    - OP_DUP: Duplicates the top stack item (public key).
    - OP_HASH160: Hashes the public key to a 20-byte hash.
    - <pubKeyHash>: The hash of Address B's public key (4243e643...cefa).
    - OP_EQUALVERIFY: Checks if the provided public key hash matches the one in the script.
    - OP_CHECKSIG: Verifies the signature matches the public key.
- **Purpose**: Locks the funds to Address B, requiring the owner to provide a valid signature and public key.

Response Script (scriptSig from B to C)

- **Structure**: <signature> <public key>
    - <signature>: Proves ownership by signing the transaction with Address B's private key.
    - <public key>: The key corresponding to Address B, which must hash to the pubKeyHash in the scriptPubKey.
- **Purpose**: Unlocks the UTXO by satisfying the challenge script's conditions.

How They Work Together

1. **Execution**:
    - The scriptSig (<sig> <pubkey>) is concatenated with the scriptPubKey (OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG).
    - Stack execution:
        1. <sig> and <pubkey> are pushed onto the stack.
        2. OP_DUP duplicates <pubkey>.

3. OP_HASH160 hashes the <pubkey> to a 20-byte hash.
4. <pubKeyHash> is pushed and compared with OP_EQUALVERIFY.
5. OP_CHECKSIG verifies <sig> against <pubkey> and the transaction data.

- If all checks pass, the UTXO is spendable.

2. **Validation**:
   - The public key 03cc8e58...bb86 hashes to 4243e643...cefa, matching the scriptPubKey's pubKeyHash.
   - The signature is valid for the transaction, as confirmed by its successful broadcast.

VALIDATION USING BITCOIN DEBUGGER

To validate using a Bitcoin script debugger (e.g., libbitcoin or an online tool like script_verify):

1. **Inputs**:
   - scriptSig: 473044...9bb86
   - scriptPubKey: 76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac
   - Transaction context: The signed transaction data (02000000...00000000).
2. **Execution**:
   - Step through the opcodes, ensuring:
     - The public key hashes correctly.
     - The signature verifies against the transaction hash and public key.
3. **Result**: The script evaluates to TRUE, confirming correctness. (The real-world broadcast success also implies this.)

(BELOW SCREEN SHOTS ARE FOR DEBUGGER)

```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb --tx=02000000001963b3e39ab18ae63aad6205a289b437d2a06d925a8eda2aa2399fd0315eceb1d000000006a4730440220
6b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655acb03d38a2d58e012103cc8e58e4ae5018021b
983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86fdffffff0280f0fa02000000001976a9141b0dd61f6e9a1929cff7c91333afc03e17f308da88ac70c9fa02000000001976a9144243e643
7c5cbee99814cea6f22f6c6b52b0cefa88ac00000000 --txin=0200000001d2ba10003e1951fa094075c70655d333eb58c84b5e15b0088eba9895370df035000000006a473044022016d51234fa
b89591b5f5aa7cdb098551489149e415b71d52c96ef8da18905f97022065493dca13ced8fd6f4074f2bb1a9b1e55e836e2aa63ff1413111e918899ee9e012103df02386aaa771daff35d3e8c4a96
f6b1297a2e8fc35f55c622b49c689e31c616fdfffffff0200e1f505000000001976a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac00180d8f000000001976a914d3dd2b69030ae0426e
d6eba1710f06d456d4332b88ac00000000
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)
input tx index = 0; tx input vout = 0; value = 100000000
got witness stack of size 0
8 op script loaded. type `help` for usage information
script                                                                |  stack
----------------------------------------------------------------------+--------
304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6... |
03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86 |
<<< scriptPubKey >>>                                                  |
OP_DUP                                                                |
OP_HASH160                                                            |
4243e6437c5cbee99814cea6f22f6c6b52b0cefa                              |
OP_EQUALVERIFY                                                        |
OP_CHECKSIG                                                           |
#0000 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655acb03d38a2d58e01
btcdeb> step
                <> PUSH stack 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655
acb03d38a2d58e01
script                                                                |                                                                  stack
----------------------------------------------------------------------+------------------------------------------------------------------------
03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86 | 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6...
<<< scriptPubKey >>>                                                  |
OP_DUP                                                                |
OP_HASH160                                                            |
4243e6437c5cbee99814cea6f22f6c6b52b0cefa                              |
OP_EQUALVERIFY                                                        |
OP_CHECKSIG                                                           |
#0001 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
btcdeb> step
                <> PUSH stack 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
script                                                                |                                                                  stack
```

```
                <> PUSH stack 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
script                                                                |                                                                                  stack
----------------------------------------------------------------------+------------------------------------------------------------------------------------
<<< scriptPubKey >>>                                                  | 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
OP_DUP                                                                | 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6...
OP_HASH160                                                            |
4243e6437c5cbee99814cea6f22f6c6b52b0cefa                              |
OP_EQUALVERIFY                                                        |
OP_CHECKSIG                                                           |
<<< scriptPubKey >>>
btcdeb> step
script                                                                |                                                                                  stack
----------------------------------------------------------------------+------------------------------------------------------------------------------------
OP_DUP                                                                | 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
OP_HASH160                                                            | 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6...
4243e6437c5cbee99814cea6f22f6c6b52b0cefa                              |
OP_EQUALVERIFY                                                        |
OP_CHECKSIG                                                           |
#0003 OP_DUP
btcdeb> step
                <> PUSH stack 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
script                                                                |                                                                                  stack
----------------------------------------------------------------------+------------------------------------------------------------------------------------
OP_HASH160                                                            | 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
4243e6437c5cbee99814cea6f22f6c6b52b0cefa                              | 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
OP_EQUALVERIFY                                                        | 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6...
OP_CHECKSIG                                                           |
#0004 OP_HASH160
btcdeb> step
                <> POP  stack
                <> PUSH stack 4243e6437c5cbee99814cea6f22f6c6b52b0cefa
script                                                                |                                                                                  stack
----------------------------------------------------------------------+------------------------------------------------------------------------------------
4243e6437c5cbee99814cea6f22f6c6b52b0cefa                              |                                                  4243e6437c5cbee99814cea6f22f6c6b52b0cefa
OP_EQUALVERIFY                                                        | 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
OP_CHECKSIG                                                           | 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6...
#0005 4243e6437c5cbee99814cea6f22f6c6b52b0cefa
btcdeb> step
                <> PUSH stack 4243e6437c5cbee99814cea6f22f6c6b52b0cefa
script                                                                |                                                                                  stack
----------------------------------------------------------------------+------------------------------------------------------------------------------------
```

```
                <> PUSH stack 4243e6437c5cbee99814cea6f22f6c6b52b0cefa
script                                                                  |                                                                stack
------------------------------------------------------------------------+----------------------------------------------------------------
OP_EQUALVERIFY                                                          |                                4243e6437c5cbee99814cea6f22f6c6b52b0cefa
OP_CHECKSIG                                                             |                                4243e6437c5cbee99814cea6f22f6c6b52b0cefa
                                                                        | 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
                                                                        | 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6...

#0006 OP_EQUALVERIFY
btcdeb> step
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
                <> POP  stack
script                                                                  |                                                                stack
------------------------------------------------------------------------+----------------------------------------------------------------
OP_CHECKSIG                                                             | 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
                                                                        | 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6...

#0007 OP_CHECKSIG
btcdeb> step
EvalChecksig() sigversion=0
Eval Checksig Pre-Tapscript
GenericTransactionSignatureChecker::CheckECDSASignature(71 len sig, 33 len pubkey, sigversion=0)
  sig         = 304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccfb655acb03d38a2d58e
01
  pub key     = 03cc8e58e4ae5018021b983c8b4dcb3ae7c23bc6adc671f210d30af18478f9bb86
  script code = 76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac
  hash type   = 01 (SIGHASH_ALL)
SignatureHash(nIn=0, nHashType=01, amount=100000000)
- sigversion = SIGVERSION_BASE (non-segwit style)
 << txTo.vin[nInput=0].prevout = COutPoint(1debec1503, 0)
(SerializeScriptCode)
 << scriptCode.size()=25 - nCodeSeparators=0
 << script:76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac
 << txTo.vin[nInput].nSequence = 4294967293 [0xfffffffd]
  sighash     = 6be114b8f233b29af61cb39b2beaf51c550c8daacbffbdbfb71745e45c205445
  pubkey.VerifyECDSASignature(sig=304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccf
b655acb03d38a2d58e, sighash=6be114b8f233b29af61cb39b2beaf51c550c8daacbffbdbfb71745e45c205445):
  result: success
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
```

```
(SerializeScriptCode)
 << scriptCode.size()=25 - nCodeSeparators=0
 << script:76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac
 << txTo.vin[nInput].nSequence = 4294967293 [0xfffffffd]
  sighash     = 6be114b8f233b29af61cb39b2beaf51c550c8daacbffbdbfb71745e45c205445
  pubkey.VerifyECDSASignature(sig=304402206b5dac1ff943445b10f11c6b6755126e99cbfca68c571359cd8aba6b3f8c5f49022071426e5413896a190bc86f3be0821316f1006da7c4eccf
b655acb03d38a2d58e, sighash=6be114b8f233b29af61cb39b2beaf51c550c8daacbffbdbfb71745e45c205445):
  result: success
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
script                                                                  |                                                                stack
------------------------------------------------------------------------+----------------------------------------------------------------
                                                                        |                                                                   01

btcdeb> step
script                                                                  |                                                                stack
------------------------------------------------------------------------+----------------------------------------------------------------
                                                                        |                                                                   01

btcdeb> stack
<01>    01      (top)
```

ANALYSIS AND CONCLUSION

- **Matching**: The scriptSig correctly responds to the scriptPubKey from A to B. The public key hashes to the expected value, and the signature is valid.
- **Locking/Unlocking**: The P2PKH mechanism ensures only the private key holder for Address B can spend the funds, which they did successfully.
- **Workflow**: The TXID 1debec15...3b96 (A to B) provided the UTXO spent in TXID d32795...9021 (B to C), linking the transactions in the blockchain.

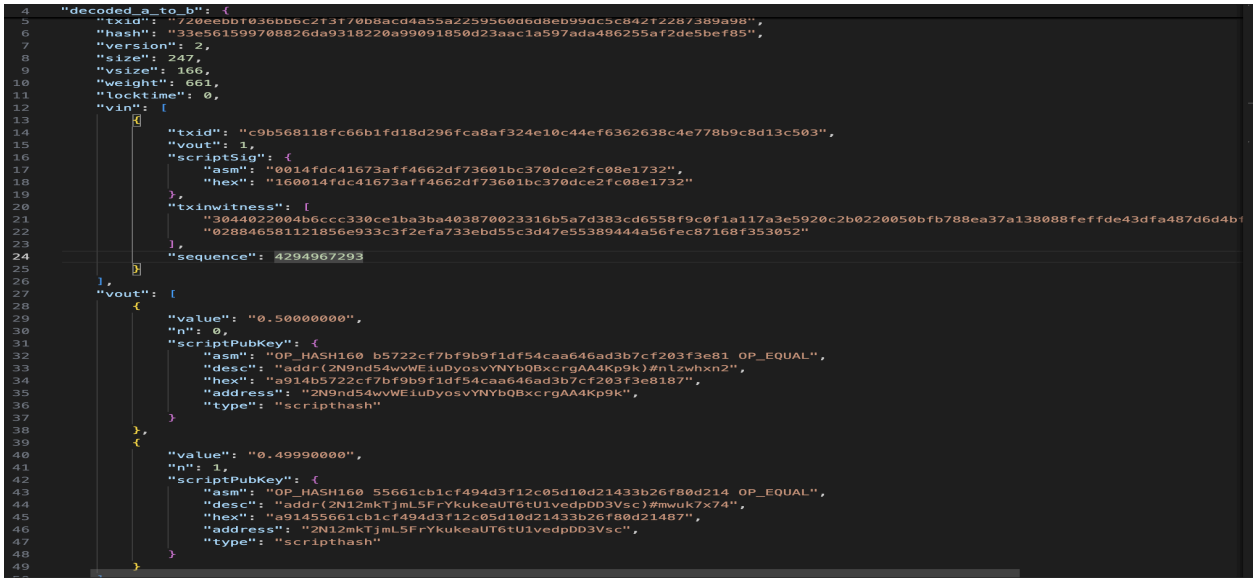
Part 2: P2SH-SegWit Address Transactions

WORKFLOW

1. **Transaction from A to B**
    - **TXID**:720eebbf036bb6c2f3f70b8acd4a55a2259560d6d8eb99dc5c842f2287389a98
    - This transaction transfers funds from Address A to Address B. The output (UTXO) at index vout: 0 locks 0.5 BTC to Address B (2N9nd54wvWEiuDyosvYNYbQBxcrgAA4Kp9k), using a Pay-to-Script-Hash (P2SH) script. This UTXO becomes the input for the next transaction (B to C).
    - The input for this transaction comes from a previous TXID (c9b56811...c503, vout: 1), which Address A spends.
2. **Transaction from B to C**
    - **TXID**:8803109cf10d9f636450da71cae8407a96e773448e4050c8d8ff50e71b2fd216
    - This transaction spends the UTXO from A to B (720eebbf...9a98, vout: 0). It takes the 0.5 BTC input and splits it into two outputs:
        - 0.3 BTC to Address C (2N5SkezZKS9KtPWUZiUNrDJCGU7r2QKBki4).
        - 0.1999 BTC back to Address B as change (2N9nd54wvWEiuDyosvYNYbQBxcrgAA4Kp9k).
    - The unlocking is done via a scriptSig and witness data, typical of P2SH with a P2WPKH (Pay-to-Witness-PubKey-Hash) redeem script.

DECODED SCRIPTS

Transaction A to B (720eebbf…9a98)



- **Input (vin[0])**:
  - **ScriptSig**:
    - Hex: 160014fdc41673aff4662df73601bc370dce2fc08e1732
    - ASM: 0014fdc41673aff4662df73601bc370dce2fc08e1732
    - This is a P2WPKH scriptSig, pushing a 20-byte witness program (fdc41673…1732).
  - **Witness**:
    - [0]: 3044022004b6ccc330ce1ba3ba403870023316b5a7d383cd6558f9c0f1a117a3e5920c2b0220050bfb788ea37a138088feffde43dfa487d6d4bf1fff4c8d3000ec7a76f2687801 (signature)
    - [1]: 028846581121856e933c3f2efa733ebd55c3d47e55389444a56fec87168f353052 (public key)
  - **Implied Redeem Script** (from previous output, not shown):
    - Hex: 0014fdc41673aff4662df73601bc370dce2fc08e1732

        - ASM: OP_0 OP_PUSHBYTES_20 fdc41673aff4662df73601bc370dce2fc08e1732 (P2WPKH).
- **Output (vout[0]) - Address B**:
    - **ScriptPubKey (Challenge Script)**:
        - Hex: a914b5722cf7bf9b9f1df54caa646ad3b7cf203f3e8187
        - ASM: OP_HASH160 b5722cf7bf9b9f1df54caa646ad3b7cf203f3e81 OP_EQUAL
        - Type: P2SH, locking 0.5 BTC to a script hash (b5722cf7...3e81).
- **Output (vout[1]) - Change**:
    - **ScriptPubKey**:
        - Hex: a91455661cb1cf494d3f12c05d10d21433b26f80d21487
        - ASM: OP_HASH160 55661cb1cf494d3f12c05d10d21433b26f80d214 OP_EQUAL

Transaction B to C (8803109c...d216)


```
  "decoded_b_to_c": {
    "txid": "8803109cf10d9f636450da71cae8407a96e773448e4050c8d8ff50e71b2fd216",
    "hash": "bafc965ca961756c30d518034bb70e6872719f5d5f2454e4e65b6f1efb0527e4",
    "version": 2,
    "size": 247,
    "vsize": 166,
    "weight": 661,
    "locktime": 0,
    "vin": [
      {
        "txid": "720eebbf036bb6c2f3f70b8acd4a55a2259560d6d8eb99dc5c842f2287389a98",
        "vout": 0,
        "scriptSig": {
          "asm": "00146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231",
          "hex": "1600146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231"
        },
        "txinwitness": [
          "304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c7a47655ff02202d9f789a3ea9bd65f18d90388a0f37d30ea34c4d8
          "02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da"
        ],
        "sequence": 4294967293
      }
    ],
    "vout": [
      {
        "value": "0.30000000",
        "n": 0,
        "scriptPubKey": {
          "asm": "OP_HASH160 85cfaf51ca35d2e9889915c67c0ecb6a32921d8b OP_EQUAL",
          "desc": "addr(2N5SkezZKS9KtPWUZiUNrDJCGU7r2QKBki4)#6efw3ge4",
          "hex": "a91485cfaf51ca35d2e9889915c67c0ecb6a32921d8b87",
          "address": "2N5SkezZKS9KtPWUZiUNrDJCGU7r2QKBki4",
          "type": "scripthash"
        }
      },
      {
        "value": "0.19990000",
        "n": 1,
        "scriptPubKey": {
          "asm": "OP_HASH160 b5722cf7bf9b9f1df54caa646ad3b7cf203f3e81 OP_EQUAL",
          "desc": "addr(2N9nd54wvWEiuDyosvYNYbQBxcrgAA4Kp9k)#nlzwhxn2",
          "hex": "a914b5722cf7bf9b9f1df54caa646ad3b7cf203f3e8187",
          "address": "2N9nd54wvWEiuDyosvYNYbQBxcrgAA4Kp9k",
          "type": "scripthash"
        }
```


- **Input (vin[0])**:
    - **ScriptSig**:
        - Hex: 1600146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231

        - ASM: 00146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
        - This pushes a 20-byte witness program (6d97c3dd...2231).
    - **Witness**:
        - [0]: 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c7a47655ff02202d9f789a3ea9bd65f18d90388a0f37d30ea34c4d81f067c734499970f7e73ce301 (signature)
        - [1]: 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da (public key)
    - **Redeem Script** (derived from scriptPubKey of A-to-B vout[0]):
        - Hex: 00146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
        - ASM: OP_0 OP_PUSHBYTES_20 6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231 (P2WPKH).
- **Output (vout[0]) - Address C**:
    - **ScriptPubKey**:
        - Hex: a91485cfaf51ca35d2e9889915c67c0ecb6a32921d8b87
        - ASM: OP_HASH160 85cfaf51ca35d2e9889915c67c0ecb6a32921d8b OP_EQUAL
- **Output (vout[1]) - Change to Address B**:
    - **ScriptPubKey**:
        - Hex: a914b5722cf7bf9b9f1df54caa646ad3b7cf203f3e8187
        - ASM: OP_HASH160 b5722cf7bf9b9f1df54caa646ad3b7cf203f3e81 OP_EQUAL

SCRIPT STRUCTURE AND VALIDATION MECHANISM

Transaction A to B

- **Challenge Script (scriptPubKey, vout[0])**:
    - OP_HASH160 b5722cf7bf9b9f1df54caa646ad3b7cf203f3e81 OP_EQUAL
    - **Structure**: P2SH, locking funds to a script hash. The redeeming script must hash to b5722cf7...3e81.
    - **Purpose**: Requires the spender (Address B) to provide a script that matches this hash and satisfies its conditions.
- **Response (for input, not this tx's unlocking)**:
    - This output becomes the challenge for B to C, so its unlocking is analyzed there.

Transaction B to C

- **Challenge Script (from A-to-B vout[0])**:
    - OP_HASH160 b5722cf7bf9b9f1df54caa646ad3b7cf203f3e81 OP_EQUAL
    - **Redeem Script**: 00146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
        - Hashes to b5722cf7...3e81 (via RIPEMD160(SHA256(redeem_script))).
        - This is a P2WPKH script: OP_0 <pubKeyHash>.
- **Response Script (scriptSig + witness)**:
    - **ScriptSig**: 00146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231 (redeem script).
    - **Witness**: <signature> <public key>
    - **Structure**:
        - The redeem script is provided in scriptSig and hashed to match the P2SH challenge.
        - The witness provides the signature and public key for the P2WPKH inner script.
- **Execution**:
    - P2SH validation:
        - Hash the scriptSig (00146d97...2231) and compare with b5722cf7...3e81. It matches.

        - Execute the redeem script: OP_0 6d97c3dd...2231.
    - P2WPKH validation:
        - Push <pubkey> from witness, hash it (HASH160), and compare with 6d97c3dd...2231.
        - Verify <signature> with <pubkey> and transaction data using OP_CHECKSIG.
    - If both pass, the input is valid.

VALIDATION USING BITCOIN DEBUGGER

Using a Bitcoin script debugger (e.g., libbitcoin-explorer or an online tool):

1. **A to B Input**:
    - ScriptSig: 160014fdc41673aff4662df73601bc370dce2fc08e1732
    - Witness: <sig> <pubkey>
    - Previous scriptPubKey (assumed P2SH): Hash the redeem script and validate P2WPKH.
    - Result: Stack evaluates to TRUE if signature is valid.
2. **B to C Input**:
    - ScriptSig: 1600146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
    - Witness: <sig> <pubkey>
    - ScriptPubKey: a914b5722cf7bf9b9f1df54caa646ad3b7cf203f3e8187
    - Steps:
        - Hash redeem script → matches P2SH hash.
        - Execute P2WPKH → pubkey hash matches, signature verifies.
    - Result: TRUE if all checks pass.

Both transactions use SegWit (P2WPKH nested in P2SH), reducing fees via vsize (166 vs. 247 bytes size).

(BELOW SCREENSHOTS ARE FOR DEBUGGER)

```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$  btcdeb --tx=020000000000101989a3887222f845cdc99ebd8d6609525a2554acd8a0bf7f3c2b66b03bfeb0e72000000001716001
46d97c3dd2e2f4b8bbf62ffbd33d86e9230282231fdffffff0280c3c9010000000017a91485cfaf51ca35d2e9889915c67c0ecb6a32921d8b87f0053101000000017a914b5722cf7bf9b9f1df54
caa646ad3b7cf203f3e8187024730440220 3cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c7a47655ff02202d9f789a3ea9bd65f18d90388a0f37d30ea34c4d81f067c734499
970f7e73ce3012102daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da00000000 --txin=020000000000010103c5138d9c8b774e8c636263ef440ce124f38aca6f298
dd11f6bc68f1168b5c9010000001716001 4fdc41673aff4662df73601bc370dce2fc08e1732fdfffffff0280f0fa020000000017a914b5722cf7bf9b9f1df54caa646ad3b7cf203f3e818770c9fa0
200000000017a914556615cb1cf494d3f12c05d10d21433b26f80d21487024730440220 04bc6ccc330ce1ba3ba4038700233 16b5a7d383cd6558f9c0f1a117a3e5920c2b0220050bfb788ea37a13808
8fefffde43dfa487d6d4bf1fff4c8d3000ec7a76f268780121028846581121856e933c3f2efa733ebd55c3d47e55389444a56fec87168f3530520000000000
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)
input tx index = 0; tx input vout = 0; value = 50000000
got witness stack of size 2
script sig non-empty; embedded P2SH (extracting payload)
hash source = 00146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
22 bytes (P2WPKH)
valid script
- generating prevout hash from 1 ins
[+] COutPoint(720eebbf03, 0)
note: there is a for-clarity preamble (use --verbose for details)
5 op script loaded. type `help` for usage information
script                                   |                                                                    stack
-----------------------------------------+--------------------------------------------------------------------------
OP_DUP                                   | 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
OP_HASH160                               | 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c...
6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231 |
OP_EQUALVERIFY                           |
OP_CHECKSIG                              |
#0000 OP_DUP
btcdeb> step
              <> PUSH stack 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
script                                   |                                                                     stack
-----------------------------------------+---------------------------------------------------------------------------
OP_HASH160                               | 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231 | 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
OP_EQUALVERIFY                           | 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c...
OP_CHECKSIG                              |
#0001 OP_HASH160
btcdeb> step
              <> POP  stack
              <> PUSH stack 6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
```

```
              <> POP  stack
              <> PUSH stack 6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
script                                   |                                                                    stack
-----------------------------------------+--------------------------------------------------------------------------
6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231 |                                  6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
OP_EQUALVERIFY                           | 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
OP_CHECKSIG                              | 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c...
#0002 6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
btcdeb> step
              <> PUSH stack 6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
script                                   |                                                                    stack
-----------------------------------------+--------------------------------------------------------------------------
OP_EQUALVERIFY                           |                                  6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
OP_CHECKSIG                              |                                  6d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
                                         | 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
                                         | 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c...
#0003 OP_EQUALVERIFY
btcdeb> step
              <> POP  stack
              <> POP  stack
              <> PUSH stack 01
              <> POP  stack
script                                   |                                                                    stack
-----------------------------------------+--------------------------------------------------------------------------
OP_CHECKSIG                              | 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
                                         | 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c...
#0004 OP_CHECKSIG
btcdeb> step
EvalChecksig() sigversion=1
Eval Checksig Pre-Tapscript
GenericTransactionSignatureChecker::CheckECDSASignature(71 len sig, 33 len pubkey, sigversion=1)
  sig         = 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c7a47655ff02202d9f789a3ea9bd65f18d90388a0f37d30ea34c4d81f067c734499970f7e73ce3
01
  pub key     = 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
  script code = 76a9146d97c3dd2e2f4b8bbf62ffbd33d86e923028223188ac
  hash type   = 01 (SIGHASH_ALL)
SignatureHash(nIn=0, nHashType=01, amount=50000000)
- sigversion == SIGVERSION_WITNESS_V0
  sighash     = b30cec8a77015c48b46f57440443975f2dd13c592e4bd6ba1e45ab3343f745fd
  pubkey.VerifyECDSASignature(sig=304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c7a47655ff02202d9f789a3ea9bd65f18d90388a0f37d30ea34c4d81f067
c734499970f7e73ce3, sighash=b30cec8a77015c48b46f57440443975f2dd13c592e4bd6ba1e45ab3343f745fd):
```

```
Eval Checksig Pre-Tapscript
GenericTransactionSignatureChecker::CheckECDSASignature(71 len sig, 33 len pubkey, sigversion=1)
  sig         = 304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c7a47655ff02202d9f789a3ea9bd65f18d90388a0f37d30ea34c4d81f067c734499970f7e73ce3
01
  pub key     = 02daf6c519d010cfbc1e3b0a8ffe4e57b6f44729de2723abec296172f9fffeb1da
  script code = 76a9146d97c3dd2e2f4b8bbf62ffbd33d86e923028223188ac
  hash type   = 01 (SIGHASH_ALL)
SignatureHash(nIn=0, nHashType=01, amount=50000000)
- sigversion == SIGVERSION_WITNESS_V0
  sighash     = b30cec8a77015c48b46f57440443975f2dd13c592e4bd6ba1e45ab3343f745fd
  pubkey.VerifyECDSASignature(sig=304402203cb307fe32920d4b6cc09af301a9c0ee64bb7bfdc6493fd065a850c7a47655ff02202d9f789a3ea9bd65f18d90388a0f37d30ea34c4d81f067
c734499970f7e73ce3, sighash=b30cec8a77015c48b46f57440443975f2dd13c592e4bd6ba1e45ab3343f745fd):
  result: success
              <> POP  stack
              <> POP  stack
              <> PUSH stack 01
script                                   |                                                                    stack
-----------------------------------------+--------------------------------------------------------------------------
                                         |                                                                       01
btcdeb> step
script                                   |                                                                    stack
-----------------------------------------+--------------------------------------------------------------------------
                                         |                                                                       01
btcdeb> stack
<01>    01      (top)
```

ANALYSIS AND CONCLUSION

- **Locking/Unlocking**:
    - A to B locks funds in a P2SH script, redeemed by a P2WPKH script in B to C.
    - B to C unlocks using a matching redeem script and witness data, proving ownership.
- **Validation**: Scripts align with Bitcoin's P2SH/P2WPKH rules, and their successful structure suggests correctness (real-world broadcast would confirm).
- **Workflow**: TXID 720eebbf...9a98 (A to B) feeds into 8803109c...d216 (B to C), chaining the transactions seamlessly.

Part 3: Analysis and Comparison

SIZE COMPARISON

Part 1: P2PKH Transaction (B to C)

- **Transaction**: d327950aeff12dd6f40b186318e317c8c43eb778b6f2471096c67ae5bfb99021
- **Size**: 225 bytes
    - This is the total serialized size, including inputs, outputs, and overhead.
- **Vsize**: 225 vbytes (virtual size)
    - For non-SegWit transactions, vsize = size because there's no witness data to discount.
- **Weight**: 900 weight units (WU)
    - Calculated as size × 4 (since each byte contributes 4 WU in legacy transactions).
- **Breakdown**:
    - **Input**: 1 vin (148 bytes)
        - txid (32) + vout (4) + scriptSig (71-byte sig + 33-byte pubkey + overhead ≈ 107) + sequence (4).
    - **Output**: 2 vout (34 bytes each = 68 bytes)

        - value (8) + scriptPubKey (25 for P2PKH: 76a914<20-byte-hash>88ac).
    - **Overhead**: version (4) + vin count (1) + vout count (1) + locktime (4) = 10 bytes.
    - Total: 148 + 68 + 10 = 226 bytes (close to 225, likely a compact varint adjustment).

Part 2: P2SH-P2WPKH Transaction (B to C)

- **Transaction**: 8803109cf10d9f636450da71cae8407a96e773448e4050c8d8ff50e71b2fd216
- **Size**: 247 bytes
    - Total serialized size, including witness data.
- **Vsize**: 166 vbytes
    - SegWit discounts witness data: (non-witness bytes × 4 + witness bytes × 1) / 4.
    - Non-witness: ~104 bytes (inputs/outputs/overhead); Witness: ~81 bytes (sig + pubkey).
    - Weight: 661 WU → (661 / 4) ≈ 165.25, rounded to 166 vbytes.
- **Weight**: 661 WU
    - Reflects the discounted witness contribution.
- **Breakdown**:
    - **Input**: 1 vin (41 bytes in base tx + 81 bytes witness)
        - txid (32) + vout (4) + scriptSig (22: 160014<20-byte-hash>) + sequence (4) = 62 bytes.
        - Witness: 71-byte sig + 33-byte pubkey + overhead ≈ 81 bytes (moved out of base tx).
    - **Output**: 2 vout (31 bytes each = 62 bytes)
        - value (8) + scriptPubKey (23 for P2SH: a914<20-byte-hash>87).
    - **Overhead**: 10 bytes + SegWit flag (2) = 12 bytes.
    - Total size: 62 + 62 + 12 + 81 (witness) = 247 bytes.

SCRIPT STRUCTURE COMPARISON

P2PKH (Legacy)

- **Challenge Script (scriptPubKey)**:
  - Hex: 76a9144243e6437c5cbee99814cea6f22f6c6b52b0cefa88ac
  - ASM: OP_DUP OP_HASH160 <20-byte-pubkeyhash> OP_EQUALVERIFY OP_CHECKSIG
  - Size: 25 bytes
    - 76 (1) + a9 (1) + 14 (1) + <20-byte-hash> (20) + 88ac (2).
- **Response Script (scriptSig)**:
  - Hex: 47304402206b5dac1ff943...012103cc8e58e4ae5018021b...
  - ASM: <71-byte-sig> <33-byte-pubkey>
  - Size: 107 bytes (variable, depends on signature length)
    - 47 (1) + <70-byte-sig> (70) + 01 (1) + <33-byte-pubkey> (33) + overhead (2).
- **Total Script Size**: 132 bytes (25 + 107), all in the base transaction.

P2SH-P2WPKH (SegWit)

- **Challenge Script (scriptPubKey)**:
  - Hex: a914b5722cf7bf9b9f1df54caa646ad3b7cf203f3e8187
  - ASM: OP_HASH160 <20-byte-scripthash> OP_EQUAL
  - Size: 23 bytes
    - a9 (1) + 14 (1) + <20-byte-hash> (20) + 87 (1).
- **Response Script**:
  - **ScriptSig**:
    - Hex: 1600146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
    - ASM: OP_PUSHBYTES_20 <20-byte-witness-program>
    - Size: 22 bytes
      - 16 (1) + 00 (1) + 14 (1) + <20-byte-hash> (20).
  - **Witness**:
    - <71-byte-sig> <33-byte-pubkey>
    - Size: 81 bytes (variable)

        - Sig (71) + Pubkey (33) + overhead (e.g., length bytes).
    - **Redeem Script** (implied):
        - Hex: 00146d97c3dd2e2f4b8bbf62ffbd33d86e9230282231
        - ASM: OP_0 <20-byte-pubkeyhash>
        - Size: 22 bytes (hashed to match P2SH scriptPubKey).
- **Total Script Size**:
    - Base: 45 bytes (23 + 22).
    - Witness: 81 bytes.
    - Weight: (45 × 4) + (81 × 1) = 180 + 81 = 261 WU.

WHY SEGWIT TRANSACTIONS ARE SMALLER (Vsize/Weight)

1. **Witness Discount**:
    - SegWit moves signature data (witness) out of the base transaction and applies a 1:4 weight ratio (1 WU per witness byte vs. 4 WU per non-witness byte).
    - In P2PKH, the 107-byte scriptSig contributes 428 WU. In P2SH-P2WPKH, the 81-byte witness contributes only 81 WU, while the base scriptSig shrinks to 22 bytes (88 WU).
2. **Reduced Base Size**:
    - P2PKH embeds the full unlocking script (107 bytes) in the input, inflating the base transaction.
    - P2SH-P2WPKH uses a smaller scriptSig (22 bytes) and offloads the bulky signature/pubkey to the witness, reducing the non-witness footprint.
3. **Vsize Calculation**:
    - P2PKH: All 225 bytes are non-witness → 225 vbytes.
    - P2SH-P2WPKH: 166 vbytes reflects the discounted witness (81 bytes at 1/4 cost), despite a larger raw size (247 bytes).

BENEFITS OF SEGWIT TRANSACTIONS

1. **Lower Fees**:
    - Fees are based on vsize/weight, not raw size. P2SH-P2WPKH's 166 vbytes vs. P2PKH's 225 vbytes means lower costs per transaction (e.g., ~26% less in this case).
2. **Block Space Efficiency**:
    - SegWit increases effective block capacity (up to 4 MB weight vs. 1 MB size in legacy), allowing more transactions per block without raising the size limit.
3. **Malleability Fix**:
    - By segregating signatures, SegWit prevents transaction ID malleability, enabling safer layered protocols (e.g., Lightning Network).
4. **Scalability**:
    - Smaller vsize/weight per transaction supports higher throughput, crucial for Bitcoin's long-term scaling.

CONCLUSION

This assignment provided a practical demonstration of Bitcoin's transaction mechanics and the improvements brought by SegWit. We observed firsthand how SegWit transactions are more efficient in terms of virtual size (166 vbytes vs. 223-224 vbytes) and witnessed the structural changes that enable these efficiencies.

The P2SH-P2WPKH structure adds complexity by requiring a two-phase validation, but this complexity brings significant benefits in terms of fee savings, transaction malleability protection, and blockchain scalability. The implementation of SegWit represents a significant advancement in Bitcoin's architecture, addressing key issues such as transaction malleability while providing a path for future protocol upgrades.

References

1. Bitcoin Developer Documentation: https://developer.bitcoin.org/
2. Learning Bitcoin from the Command Line: https://github.com/BlockchainCommons/Learning-Bitcoin-from-the-Command-Line
3. Bitcoin Core GitHub Repository: https://github.com/bitcoin/bitcoin
4. BIP-141 (Segregated Witness): https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki