

CS 216: Bitcoin Scripting Assignment Report

Part 1: Legacy Address Transactions

1. Transaction Workflow:

The script executes a sequence of Bitcoin transactions on a local Bitcoin Core node using the `AuthServiceProxy` RPC interface. The primary workflow follows:

1. Wallet and Address Setup:

- Load a wallet (`mywallet`)
- Generate three legacy addresses: `Legacy A`, `Legacy B` and `Legacy C`.

2. Mining and Funding:

- Mine 201 blocks to `Legacy A`, ensuring sufficient balance.
- Send 10 BTC from `Legacy A` to itself, generating an unspent transaction output (UTXO) for further use.

3. Transaction 1: Sending BTC from A to B:

- Fetch the UTXO created from the funding transaction.
- Construct a raw transaction where:
 - 5 BTC is sent to `Legacy B`.
 - Change (original UTXO minus sent amount and fee) is returned to `Legacy A`.
- Sign and broadcast the transaction.

4. Transaction 2: Sending BTC from B to C

- Fetch the newly created UTXO from the A to B transaction.
- Construct a raw transaction where:
 - 2.5 BTC is sent to `Legacy C`.
 - Remaining balance (minus fee) is returned to `Legacy B`.
- Sign and broadcast the transaction.

The transaction sending BTC

2. Decoded Scripts for Both Transactions

1. Transaction 1 (A → B):

```
Decoded signed transaction {'txid': '07055bf40b43a6f859c4de9c4af7d82bb9b0c6a69173ec44b01556dfcf5411f8', 'hash': '07055bf40b43a6f859c4de9c4af7d82bb9b0c6a69173ec44b01556dfcf5411f8', 'version': 2, 'size': 225, 'vsize': 225, 'weight': 900, 'locktime': 0, 'vin': [{'txid': '5e53be790ad26e97f518a74b02e98520617c0b8b5b0fd00e483fa32edad10489', 'vout': 0, 'scriptSig': {'asm': '3044022052d5fcef3ea8221eec24d69e63ff941618fe7cc06a1bfa1d7765ca42245d47f00220119ca14a164b101f58fdd480a4e844199e55c12a35c2f3c2cc02e19a1943e274[ALL] 0388623bbefb96515960891573c2ca44b3a1410c352651225f9242525dd5f85e31', 'hex': '473044022052d5fcef3ea8221eec24d69e63ff941618fe7cc06a1bfa1d7765ca42245d47f00220119ca14a164b101f58fdd480a4e844199e55c12a35c2f3c2cc02e19a1943e27401210388623bbefb96515960891573c2ca44b3a1410c352651225f9242525dd5f85e31'}, 'sequence': 4294967293}], 'vout': [{'value': Decimal('5.00000000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 a1700221e2435191a355d390d6064377326f3753 OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(mvEZGQxaaDEr8Eb5W5Ryu9F44PBkcAdvRD)#z82zwgdf', 'hex': '76a914a1700221e2435191a355d390d6064377326f375388ac', 'address': 'mvEZGQxaaDEr8Eb5W5Ryu9F44PBkcAdvRD', 'type': 'pubkeyhash'}}, {'value': Decimal('4.99990000'), 'n': 1, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 44f71eee441438c6f4efc4ef50fe1e55d2f77b4c OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(mmocHmW7DdxpgUm5UKgEroQLKEkZJDxuah)#918f84h9', 'hex': '76a91444f71eee441438c6f4efc4ef50fe1e55d2f77b4c88ac', 'address': 'mmocHmW7DdxpgUm5UKgEroQLKEkZJDxuah', 'type': 'pubkeyhash'}}]}
```

2. Transaction 2 (B → C):

```
Decoded signed transaction: {'txid': '7284bfc0465c70cf3d0788d8a9d46439e6f154b6fae266350ace91b2c665abb5', 'hash': '7284bfc0465c70cf3d0788d8a9d46439e6f154b6fae266350ace91b2c665abb5', 'version': 2, 'size': 225, 'vsize': 225, 'weight': 900, 'locktime': 0, 'vin': [{'txid': '07055bf40b43a6f859c4de9c4af7d82bb9b0c6a69173ec44b01556dfc5411f8', 'vout': 0, 'scriptSig': {'asm': '30440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd125b6e1b902200d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe[ALL] 02c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc', 'hex': '4730440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd125b6e1b902200d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe012102c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc', 'sequence': 4294967293}], 'vout': [{'value': Decimal('2.50000000'), 'n': 0, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 3442621ffa9e8622f08a3da98255d54f7ddb9dc2 OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(mkHGxRR7uXtg7zaYvDnz3X1xVUfchcFQyD)#yxemwadj', 'hex': '76a9143442621ffa9e8622f08a3da98255d54f7ddb9dc288ac', 'address': 'mKH6XR7uXtg7zaYvDnz3X1xVUfchcFQyD', 'type': 'pubkeyhash'}}, {'value': Decimal('2.49990000'), 'n': 1, 'scriptPubKey': {'asm': 'OP_DUP OP_HASH160 a1700221e2435191a355d390d6064377326f3753 OP_EQUALVERIFY OP_CHECKSIG', 'desc': 'addr(mvEZGQxaaDer8Eb5W5Ryu9F44PBKcAdvRD)#z822zwgdf', 'hex': '76a914a1700221e2435191a355d390d6064377326f375388ac', 'address': 'mvEZGQxaaDer8Eb5W5Ryu9F44PBKcAdvRD', 'type': 'pubkeyhash'}}]}
```

3. Challenge and Response Script Analysis:

Each transaction uses P2PKH scripts, which involve a challenge (locking script) and a response (unlocking script).

- 1. Locking Script (Challenge):** Ensures only the recipient can spend the output.
- 2. Unlocking Script (Response):** Provides a valid signature and public key.
- 3. Validation Process:**

- Bitcoin nodes execute the unlocking script first, pushing the provided signature and public key onto the stack.
- Then, the locking script is executed to verify ownership by checking that the provided public key hash matches and that the signature is valid.

4. Validation Using Bitcoin Debugger:

```
gdb> (gdb) run --txid=7284bfc0465c70cf3d0788d8a9d46439e6f154b6fae266350ace91b2c665abb5 --tx=0200000001f81154cfd5615b044ec7391a6c6b0b92b08f74bdcdec439f8a6430bf45b807000000006a73040220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd125b6e1b902200d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe[ALL] 02c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc --txin=0200000001b904d1da2ea33f480ed00f9b0b07c512085e9824ba718f5976ed2a795b535e00000000a7304022052d5fcef3ea821e0c24d696e3ff941618f7c0e61fa1d7765c4a2245d07f00220119ca14a164b101f58f6d048a6e44199e55c12a35c2f3c2c02e19a1943e7701210308623bbefb96515960891573c2ca4b3a1410c352651225f924252d5f8e31d7fffff020005cd1d000000001976a914a1700221e2435191a355d390d6064377326f3753 OP_EQUALVERIFY OP_CHECKSIG --txout=0200000001b904d1da2ea33f480ed00f9b0b07c512085e9824ba718f5976ed2a795b535e00000000a7304022052d5fcef3ea821e0c24d696e3ff941618f7c0e61fa1d7765c4a2245d07f00220119ca14a164b101f58f6d048a6e44199e55c12a35c2f3c2c02e19a1943e7701210308623bbefb96515960891573c2ca4b3a1410c352651225f924252d5f8e31d7fffff020005cd1d000000001976a914a1700221e2435191a355d390d6064377326f3753 OP_EQUALVERIFY OP_CHECKSIG --txoutindex=0 --txoutvout=0 --txoutvalue=580000000
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 = 580000000
got witness stack of size 0
$ op script loaded. type 'help' for usage information

script                                     stack
-----
30440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd...
82c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<< scriptPubKey >>>
OP_DUP
OP_HASH160
a1700221e2435191a355d390d6064377326f3753
OP_EQUALVERIFY
OP_CHECKSIG
#0000 30440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd125b6e1b902200d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
btcdeb> step
script                                     stack
-----
82c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
30440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd...
<< scriptPubKey >>>
OP_DUP
OP_HASH160
a1700221e2435191a355d390d6064377326f3753
OP_EQUALVERIFY
OP_CHECKSIG
#0001 82c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
btcdeb> print
#0000 30440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd125b6e1b902200d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
-> #0001 82c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<<< scriptPubKey >>>
#0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
btcdeb> step
script                                     stack
-----
<<< scriptPubKey >>>
#0000 30440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd...
#0001 82c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
OP_DUP
OP_HASH160
a1700221e2435191a355d390d6064377326f3753
OP_EQUALVERIFY
OP_CHECKSIG
<<< scriptPubKey >>>
btcdeb> print
#0000 30440220680320259492e95963a06c2c465afd55f800760cd43034fdb4ce2cd125b6e1b902200d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
#0001 82c43cb4973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
-> <<< scriptPubKey >>>
#0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
```

```

btcd> step
script
-----|-----
OP_DUP                                | 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
OP_HASH160                           | 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
a1700221e2435191a355d390d6064377326f3753
OP_EQUALVERIFY                       |
OP_CHECKSIG                           |
#0003 OP_DUP
btcd> print
#0000 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
#0001 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<<< scriptPubKey >>>
-> #0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
btcd> step
<> PUSH stack 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
script
-----|-----
OP_HASH160                                | 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
a1700221e2435191a355d390d6064377326f3753 | 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
OP_EQUALVERIFY                           | 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd...
OP_CHECKSIG                              |
#0004 OP_HASH160
btcd> print
#0000 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
#0001 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<<< scriptPubKey >>>
-> #0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
btcd> step
<> POP stack
<> PUSH stack a1700221e2435191a355d390d6064377326f3753
script
-----|-----
a1700221e2435191a355d390d6064377326f3753 | a1700221e2435191a355d390d6064377326f3753
OP_EQUALVERIFY                           | 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
OP_CHECKSIG                              | 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd...
#0005 a1700221e2435191a355d390d6064377326f3753
btcd> print
#0000 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
#0001 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<<< scriptPubKey >>>
-> #0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
btcd> step
<> PUSH stack a1700221e2435191a355d390d6064377326f3753
script
-----|-----
OP_EQUALVERIFY                                | a1700221e2435191a355d390d6064377326f3753
OP_CHECKSIG                                  | 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
#0006 OP_EQUALVERIFY                       | 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd...

```

```

#0007 OP_EQUALVERIFY
btcd> print
#0000 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
#0001 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<<< scriptPubKey >>>
-> #0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
btcd> step
<> POP stack
<> POP stack
<> PUSH stack 01
<> POP stack
script
-----|-----
OP_CHECKSIG                                | 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
#0007 OP_CHECKSIG                         | 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd...
btcd> print
#0000 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
#0001 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<<< scriptPubKey >>>
-> #0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
btcd> step
EvalChecksig() sigversion=0
EvalChecksig ffe=1, script
GenericTransactionSignatureChecker: {CheckECDSAASignature(71 len sig, 33 len pubkey, sigversion=0)}
sig      = 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
pub key  = 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
script code = 76a914a1700221e2435191a355d390d6064377326f375388ac
hash type = 01 (SIGHASH_ALL)
SignatureHash(in=0, mhashType=01, amount=800000000)
- sigversion = SIGHASH_ALL, BASE (non-legacy style)
<< tx.to.vin[ninput=0].prevout = COutPoint(07055bf40b, 0)
(SerializeScriptCode)
<< scriptCode size=025 - mCodeSeparators=8
<< script: 76a914a1700221e2435191a355d390d6064377326f375388ac
<< tx.to.vin[ninput].nSequence = 4294967293 (8xffffffff)
sighash  = 10f562b764a0181d0842528624d70b4b81241e0f643b6da36f8
pubkey.VerifyECDSAASignature(sig=30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01, sighash=10f562b764a0181d0842528624d70b4b81241e0f643b6da36f8
2259252195e8):
result: success
<> POP stack
<> POP stack
<> PUSH stack 01
script
-----|-----
01
btcd> print
#0000 30440226680320259492e95963a86c2c465afd55f800760cd43034fdb4bce2cd125b6e1b902208d5ec1bf7510781a8af365dc898624df99fffe57d5e2a7296936ac70f2c07ffe01
#0001 02c43cb973fc4e84a8292552717f603f337d994396d2584aa5ac815fbcfb398dc
<<< scriptPubKey >>>
-> #0003 OP_DUP
#0004 OP_HASH160
#0005 a1700221e2435191a355d390d6064377326f3753
#0006 OP_EQUALVERIFY
#0007 OP_CHECKSIG
btcd> |

```

Part 2: P2SH-SegWit Address Transactions

1. Transaction Workflow and Execution

1. Initial Funding of Address A:

- A Bitcoin address (Address A) is generated.
- 201 blocks are mined to this address, ensuring the funds mature and become spendable.
- The wallet balance is retrieved.

- A transaction of 10 BTC is sent to Address A, generating a unique transaction ID.
- The unspent transaction output (UTXO) associated with Address A is then identified.

2. Transaction from Address A to Address B:

- A UTXO from `txid_fund` is selected as input for the new transaction.
- A transfer of 5 BTC is initiated from Address A to Address B, with a small transaction fee deducted.
- A raw transaction is then formulated using `createrawtransaction()`.
- The raw transaction is then signed using `signrawtransactionwithwallet()`.
- The signed transaction is broadcasted, generating a new transaction ID.
- The decoded transaction confirms that Address B successfully received 5 BTC.

3. Transaction from Address B to Address C:

- The UTXO from `txid` is used as an input for another transaction.
- 2.5 BTC is sent from Address B to Address C.
- A new raw transaction is generated, signed, and broadcasted following the same process.
- The decoded transaction confirms that Address C received 2.5 BTC.

2. Decoded Transaction Scripts:

1. Transaction 1 (A → B):

```
Decoded signed transaction {'txid': 'cc177bf62f097bdf7b10d650cb4b067270f510cb4700412329b110493b8ec5d0', 'hash': 'aa1bdf3bb1535b31ba0d74d1b3bfd9e5f4a5bcdcd93ea15d14c0962228882c63', 'version': 2, 'size': 222, 'vsize': 141, 'weight': 561, 'locktime': 0, 'vin': [{'txid': 'dd22775ca9c225c5718634b563aa085153482af3ab668fbad43858b3c089f7d1', 'vout': 0, 'scriptSig': {'asm': '', 'hex': ''}, 'txinwitness': ['304402206034c34147d2e570d63c15d362b4716b273f945468de0a77bf5332ba1618c0c502205b5946c84eced99553ee707e9ccc105c975ecc306a5ae468ba2797d88d992dc001', '0331455f29c29ed4aa136bb6cd3a489a1f7717ea52bbc485a5b5947251b49b811d'], 'sequence': 4294967293}], 'vout': [{'value': Decimal('5.00000000'), 'n': 0, 'scriptPubKey': {'asm': '0 ee04e98bb1428ff52b4c1429dc33fe90acb57795', 'desc': 'addr(bcrt1qaczwznza3g28l226vzs5acv17jzkt2au4dnhkp4)#8sprmq6g', 'hex': '0014ee04e98bb1428ff52b4c1429dc33fe90acb57795', 'address': 'bcrt1qaczwznza3g28l226vzs5acv17jzkt2au4dnhkp4', 'type': 'witness_v0_keyhash'}}, {'value': Decimal('4.99990000'), 'n': 1, 'scriptPubKey': {'asm': '0 49341f0a16a6ba5e638104d02d3c08543340cba9', 'desc': 'addr(bcrt1qfy6p7zsk56a9ucupqngz60qg2se5pjafmkf5zg)#9d4y8wp', 'hex': '001449341f0a16a6ba5e638104d02d3c08543340cba9', 'address': 'bcrt1qfy6p7zsk56a9ucupqngz60qg2se5pjafmkf5zg', 'type': 'witness_v0_keyhash'}}]}
```

2. Transaction 2 (B → C):

```
Decoded signed transaction: {'txid': '088d4afb3ba715b3212982c6b4497c36df6e38a6d5c32d66485f7afa6cea784f', 'hash': 'ab42ae27aeb731aac0a9df501eb19e9cbf4d19d09748964825ceb5c2861ac1aa', 'version': 2, 'size': 222, 'vsize': 141, 'weight': 561, 'locktime': 0, 'vin': [{'txid': 'cc177bf62f097bdf7b10d650cb4b067270f510cb4700412329b110493b8ec5d0', 'vout': 0, 'scriptSig': {'asm': '', 'hex': ''}, 'txinwitness': ['304402202bafd45411f65b8a3c5fe00d70d22cefe25198e46e25e1a19a2245d60f0e2e3f022060a1d370d2449b5720e3fbdcd9a50a31522fb95930612f731810e0926d7d785801', '02785ac0e7ca86dfe495b46ffeb7ba382b650d890b464eb533ef57b5b3ea31d016'], 'sequence': 4294967293}], 'vout': [{'value': Decimal('2.50000000'), 'n': 0, 'scriptPubKey': {'asm': '0 1a74f3286f43cdd4fbfb0b293b3e68e822c9c670', 'desc': 'addr(bcrt1qrf60x2r0g0xaf7lmpv5nk0nga3vn3ns85ywkq)#ehgl50t9', 'hex': '00141a74f3286f43cdd4fbfb0b293b3e68e822c9c670', 'address': 'bcrt1qrf60x2r0g0xaf7lmpv5nk0nga3vn3ns85ywkq', 'type': 'witness_v0_keyhash'}}, {'value': Decimal('2.49990000'), 'n': 1, 'scriptPubKey': {'asm': '0 ee04e98bb1428ff52b4c1429dc33fe90acb57795', 'desc': 'addr(bcrt1qaczwznza3g28l226vzs5acv17jzkt2au4dnhkp4)#8sprmq6g', 'hex': '0014ee04e98bb1428ff52b4c1429dc33fe90acb57795', 'address': 'bcrt1qaczwznza3g28l226vzs5acv17jzkt2au4dnhkp4', 'type': 'witness_v0_keyhash'}}]}
```

3. Analysis of Challenge and Response Scripts:

Bitcoin Script Execution

Bitcoin transactions use a challenge-response mechanism through scripting.

- 1. Locking Script (scriptPubKey):** This script defines the conditions required to unlock the UTXO. A typical Pay-to-Public-Key-Hash (P2PKH) script follows the structure:

- OP_DUP: Duplicates the public key on the stack.
- OP_HASH160: Computes the hash of the public key.
- OP_EQUALVERIFY: Verifies that the provided public key hash matches the expected hash.
- OP_CHECKSIG: Validates the signature against the public key.

2. Unlocking Script (scriptSig): This script provides the necessary proof to meet the locking script conditions:

- The digital signature is generated using the sender's private key and proves ownership of the UTXO.

Transaction Validation Process:

3. The unlocking script (`scriptSig`) executes first, placing the signature and public key onto the stack.
4. The locking script (`scriptPubKey`) runs, verifying the public key's hash and the authenticity of the digital signature.
5. If all conditions hold, the transaction is deemed valid and accepted into the blockchain.

4. Validation using Bitcoin Debugger:

[illegible]

```

btcd> print
#000 OP_DUP
#001 OP_HASH160
#002 ee94e98bb1428ff52bdc1429dc33fe90ac57795
-> #003 OP_EQUALVERIFY
#004 OP_CHECKSIG
btcd> step
<> POP stack
<> POP stack
<> PUSH stack #1
<> POP stack

script
-----|-----
OP_CHECKSIG | 02785ac0e7ca86dfe495b46ffeb7ba32b5d8d99b464eb533ef57b5b3ea31d016
          | 304402202baf0d5411f65b8a3c5fe08d70d22cfe23198e46e25e1a19a2245d...
#004 OP_CHECKSIG
btcd> print
#000 OP_DUP
#001 OP_HASH160
#002 ee94e98bb1428ff52bdc1429dc33fe90ac57795
#003 OP_EQUALVERIFY
-> #004 OP_CHECKSIG
btcd> step
EvalCheckSig() sigversion=1
EvalCheckSig Pre-Script
GenericTransactionSignatureChecker::CheckECDSA(Signature(71 len sig, 33 len pubkey, sigversion=1)
sig      = 304402202baf0d5411f65b8a3c5fe08d70d22cfe23198e46e25e1a19a2245d60fe2e3f022060a1d370d2449b5720e3fbdcd9a50a31522fb95930612f731810e0926d7d785801
pub key  = 02785ac0e7ca86dfe495b46ffeb7ba32b5d8d99b464eb533ef57b5b3ea31d016
script code = 76a914ee0e98bb1428ff52bdc1429dc33fe90ac5779588ac
hash type  = 01 (SIGHASH_ALL)
SignatureHash(nIn=0, mhashType=01, amount=500000000)
- sigversion == SIGVERSION_WITNESS_V0
sighash = 7e67d1cb27853a81c3497f0254f2d6d2d738fdbadabc0d6dace63871cad9e5f
pubkey.VerifyECDSA(Signature(sig=304402202baf0d5411f65b8a3c5fe08d70d22cfe23198e46e25e1a19a2245d60fe2e3f022060a1d370d2449b5720e3fbdcd9a50a31522fb95930612f731810e0926d7d7858, sighash=7e67d1cb27853a81c3497f0254f2d6d2d738fdbadabc0d6dace63871cad9e5f)
result: success
<> POP stack
<> POP stack
<> PUSH stack #1

script
-----|-----
          | 01
btcd> print
#000 OP_DUP
#001 OP_HASH160
#002 ee94e98bb1428ff52bdc1429dc33fe90ac57795
#003 OP_EQUALVERIFY
#004 OP_CHECKSIG
btcd> |

```

Step 3: Comparison of Part 1 and Part 2:

1. Address Format:

- Legacy:** Uses P2PKH (Pay-to-PubKey-Hash) addresses, starting with `m` (for regtest). Eg: `mmocHmW7DdxpgUm5UKgEroQLKEkZJDxuah`
- SegWit:** Uses Bech32 (P2WPKH) addresses, starting with `bcrt1`. Eg: `bcrt1qfy6p7zsk56a9ucupqngz60qg2se5pjafmkf5zg`

2. Transaction Size & Weight:

- Legacy:**
 - Size:** Larger in size (225 bytes - `vsize`). This is because it includes full signatures inside `scriptSig`.
 - Weight:** Higher weight (900). This is because of inefficient signature handling.
- SegWit:**
 - Size:** Smaller in size (141 bytes – `vsize`). This is because signatures are moved to `witness` data.
 - Weight:** Reduced weight (561), allowing more transactions per block.

3. Transaction Structure:

- Legacy:**
 - Uses `scriptSig` in the input for unlocking funds, including the full signature and public key.
- SegWit:**
 - Uses `witness` data instead of `scriptSig`, making transactions more compact.

SegWit transactions are smaller primarily because they move the signature data (witness data) outside the main transaction structure. Here's how it works:

1. Signature Data is Moved to the Witness Field:

- **Legacy transactions:** Signatures are included in the scriptSig field of each input. This makes the transaction larger because every input must include a full signature.
- **SegWit transactions:** Signatures are moved to the witness section, which is not included in the transaction ID calculation. This reduces the size of the main transaction.

2. Witness Data is Discounted:

- Bitcoin blocks are limited by weight units instead of just size in bytes.
- Witness data is discounted by a factor of 4, meaning 1 byte of witness data counts as only 1/4th of a regular byte.
- As a result, SegWit transactions are smaller in terms of vsize (virtual size) when compared to Legacy transactions.

Benefits of SegWit Transactions:

1. **Lower Transaction Fees:** Since SegWit transactions have a smaller virtual size, they pay lower fees compared to legacy transactions.

2. More Transactions per Block:

- a. Bitcoin blocks have a maximum weight limit of 4 million weight units instead of a strict byte limit.
- b. Since witness data is discounted, SegWit transactions take up less weight, meaning more transactions can fit in a single block.
- c. This increases Bitcoin's transaction throughput without increasing the block size.

3. Fixes Transaction Malleability:

- a. In Legacy transactions, changing the scriptSig (even in a trivial way) changes the transaction ID (TXID).
- b. In SegWit, the signature data (witness) is excluded from the TXID calculation, meaning TXIDs remain stable even if the witness data is modified.
- c. This fix enables second-layer solutions like the Lightning Network, which relies on predictable TXIDs.