CS 216: INTRODUCTION TO BLOCKCHAIN

08TH MARCH 2024

ASSIGNMENT 3 : BITCOIN SCRIPTING

TEAM: MINEOVERMATTER

230001003	ABHINAV BITRAGUNTA
230001006	AMAN GUPTA
230001072	SRINIDHI SAI BOORGU

INTRODUCTION

This report describes our implementation and analysis of Bitcoin transactions using Legacy (P2PKH) and SegWit (P2SH-P2WPKH) address formats. We've written Python scripts to interact with `bitcoind`, create transactions, analyze and debug the Bitcoin scripts involved in these transactions.

BITCOIN-CORE CONFIGURATION

```
[regtest]
regtest=1
server=1
rpcuser=<rpc_credential>
rpcpassword=<rpc_credential>
rpcallowip=127.0.0.1
rpcport=18443
txindex=1

paytxfee=0.0001
fallbackfee=0.0002
mintxfee=0.00001
txconfirmtarget=6
```

Regtest configurations

PART-1: LEGACY ADDRESS TRANSACTIONS (P2PKH)

WORKFLOW

1. We generated three legacy addresses: A, B, and C
2. Funded address A using the `sendtoaddress` command
3. Created a transaction from A to B using `createrawtransaction`
4. Decoded the transaction to examine the locking script (ScriptPubKey) for B
5. Signed and broadcast the transaction, generating txid: <txid_a_to_b>
6. Retrieved this unspent transaction output (UTXO) for address B
7. Created a transaction from B to C using the UTXO from the previous transaction
8. Decoded, signed, and broadcast this transaction, generating txid: <txid_b_to_c>

SCRIPT ANALYSIS

Transaction A to B:

The ScriptPubKey (locking script) for address B follows the P2PKH pattern:

```
OP_DUP OP_HASH160 <pubkey hash> OP_EQUALVERIFY OP_CHECKSIG
```

The script does the following:

1. Duplicates the provided public key
2. Hashes it using HASH160 (RIPEMD160(SHA256))
3. Checks if the hash matches the stored pubkey hash
4. Verifies the signature against the public key

```
PS C:\Program Files\Bitcoin\daemon> .\bitcoin-cli.exe -regtest decoderawtransaction (.\bitcoin-cli.exe -regtest getrawtransaction "07b3c2c3e791e11c3faffb033
3e207bb016633cf4f727621e56874d1e0f48c9f")
{
  "txid": "07b3c2c3e791e11c3faffb0333e207bb016633cf4f727621e56874d1e0f48c9f",
  "hash": "07b3c2c3e791e11c3faffb0333e207bb016633cf4f727621e56874d1e0f48c9f",
  "version": 2,
  "size": 191,
  "vsize": 191,
  "weight": 764,
  "locktime": 0,
  "vin": [
    {
      "txid": "82a4a2058873e4045e0ba3005e05f304dc129328965f9f0e5743ae92c2436251",
      "vout": 0,
      "scriptSig": {
        "asm": "30440220461807 41b0e42bb09967e936b2c83c979c8faf6337cfe8767313f78a214df38902204bb3c52af05de8b1f11c6bae40589218b0630a90d231356bdc70e648ac7bb9e8
[ALL] 023df1d2022a913313ba8d742e96f08d00d2ca7fd14ae2fd4599aa0f6de6d5612d",
        "hex": "4730440220461807 41b0e42bb09967e936b2c83c979c8faf6337cfe8767313f78a214df38902204bb3c52af05de8b1f11c6bae40589218b0630a90d231356bdc70e648ac7bb9
e80121023df1d2022a913313ba8d742e96f08d00d2ca7fd14ae2fd4599aa0f6de6d5612d"
      },
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": 0.99990000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_DUP OP_HASH160 0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc OP_EQUALVERIFY OP_CHECKSIG",
        "desc": "addr(mfnzo7byi9R76pf4Ct1eUH5cgf9QSA6qnv)#5d7z884z",
        "hex": "76a9140308be3f7ded2074b14c477fa18ec0ecfe3e3dfc88ac",
        "address": "mfnzo7byi9R76pf4Ct1eUH5cgf9QSA6qnv",
        "type": "pubkeyhash"
      }
    }
  ]
}
```

Transaction from A to B

Transaction B to C:

The ScriptSig (unlocking script) for spending from address B:

```
<signature> <public key>
```

This unlocking script provides:

1. The signature for the transaction
2. The public key corresponding to address B

```
PS C:\Program Files\Bitcoin\daemon> .\bitcoin-cli.exe -regtest decoderawtransaction (.\bitcoin-cli.exe -regtest getrawtransaction "bae2c3576128e35c649ba3053
8b9e24aecf300aad7063a0e230a344fd35ffa23")
{
  "txid": "bae2c3576128e35c649ba30538b9e24aecf300aad7063a0e230a344fd35ffa23",
  "hash": "bae2c3576128e35c649ba30538b9e24aecf300aad7063a0e230a344fd35ffa23",
  "version": 2,
  "size": 191,
  "vsize": 191,
  "weight": 764,
  "locktime": 0,
  "vin": [
    {
      "txid": "07b3c2c3e791e11c3faffb0333e207bb016633cf4f727621e56874d1e0f48c9f",
      "vout": 0,
      "scriptSig": {
        "asm": "3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a937073793f1bfb02205d4441054c0f984cf795deac8e9034e68b20cc323cd3332d45f00666811f8315
[ALL] 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb",
        "hex": "473044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a937073793f1bfb02205d4441054c0f984cf795deac8e9034e68b20cc323cd3332d45f00666811f83
15012102a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb"
      },
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": 0.99980000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_DUP OP_HASH160 0ec0cbaef7e11b93bfb59d898b5f611ff227b8a1 OP_EQUALVERIFY OP_CHECKSIG",
        "desc": "addr(mgrxiR3r336phqmN2EGuTtMCF25PJsBK4u)#cfwl59e7",
        "hex": "76a9140ec0cbaef7e11b93bfb59d898b5f611ff227b8a188ac",
        "address": "mgrxiR3r336phqmN2EGuTtMCF25PJsBK4u",
        "type": "pubkeyhash"
      }
    }
  ]
}
```

Transaction from B to C

When executed with the locking script from the previous transaction, it forms a complete validation script:

`<signature> <public key> OP_DUP OP_HASH160 <pubkey hash> OP_EQUALVERIFY OP_CHECKSIG`

Challenge-Response Validation using `btcdeb`*:*

```
cse@cse-HP-ProDesk-600-G5-PCI-MT:~/Desktop/btcdeb/btcdeb$ btcdeb --verbose '[3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a937073793f1bfb02205d4441054c0f984cf795deac8e9034e68b20cc323cd3332d45f00666811f8315 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb OP_DUP OP_HASH160 0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc OP_EQUALVERIFY OP_CHECKSIG]'
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
7 op script loaded. type `help` for usage information
```

```
script                                                             |  stack
-------------------------------------------------------------------+--------
3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a93707... |
02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb |
OP_DUP                                                             |
OP_HASH160                                                         |
0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc                           |
OP_EQUALVERIFY                                                     |
OP_CHECKSIG                                                        |
#0000 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a937073793f1bfb02205d4441054c0f984cf795deac8e9034e68b20cc323cd3332d45f00666811f8315
```

Loading the Script

1. Push signature and public key onto the stack

```
btcdeb> step
                <> PUSH stack 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a937073793f1bfb02205d4441054c0f984cf795deac8e9034e68b20cc323cd3332d45f00666811f8315
script                                                             |                                                              stack
-------------------------------------------------------------------+-------------------------------------------------------------------
02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb | 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a93707...
OP_DUP                                                             |
OP_HASH160                                                         |
0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc                           |
OP_EQUALVERIFY                                                     |
OP_CHECKSIG                                                        |
#0001 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
```

```
btcdeb> step
                <> PUSH stack 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
script                                                             |                                                              stack
-------------------------------------------------------------------+-------------------------------------------------------------------
OP_DUP                                                             | 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
OP_HASH160                                                         | 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a93707...
0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc                           |
OP_EQUALVERIFY                                                     |
OP_CHECKSIG                                                        |
#0002 OP_DUP
```

2. OP_DUP duplicates the public key

```
btcdeb> step
                <> PUSH stack 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
script                                                             |                                                              stack
-------------------------------------------------------------------+-------------------------------------------------------------------
OP_HASH160                                                         | 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc                           | 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
OP_EQUALVERIFY                                                     | 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a93707...
OP_CHECKSIG                                                        |
#0003 OP_HASH160
```

3. `OP_HASH160` hashes the public key

```
btcdeb> step
                <> POP  stack
                <> PUSH stack 0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc
script                                                             |                                                              stack
-------------------------------------------------------------------+-------------------------------------------------------------------
0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc                           |                           0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc
OP_EQUALVERIFY                                                     | 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
OP_CHECKSIG                                                        | 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a93707...
#0004 0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc
```

4. The hash is compared with the stored pubkey hash

```
btcdeb> step
                <> PUSH stack 0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc
script                                                             |                                                              stack
-------------------------------------------------------------------+-------------------------------------------------------------------
OP_EQUALVERIFY                                                     |                           0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc
OP_CHECKSIG                                                        |                           0308be3f7ded2074b14c477fa18ec0ecfe3e3dfc
                                                                   | 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
                                                                   | 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a93707...
#0005 OP_EQUALVERIFY
```

5. OP_EQUALVERIFY ensures they match

```
btcdeb> step
                <> POP  stack
                <> POP  stack
                <> PUSH stack 01
                <> POP  stack
script                                                             |                                                              stack
-------------------------------------------------------------------+-------------------------------------------------------------------
OP_CHECKSIG                                                        | 02a6b47f2ea4d66df2b14b9878c53fe58fec9724500f74390ec45b43994c374bfb
                                                                   | 3044022073fdb36a203bc0eb4ffa8d2545f26cb4aeec78ff31ae92d79a93707...
#0006 OP_CHECKSIG
```

6. `OP_CHECKSIG` verifies the signature against the public key and pushes a 01 onto the stack if it is valid*.

*We have tried to validate the signature using `btcdeb` but the debugger doesn't recognise the signature appropriately. Upon looking through the GitHub tutorial page, we found the following statement: "Unfortunately this checking may or may not be working at any point due to vagaries of the Bitcoin Core and `btcdeb` code."

PART 2: SEGWIT ADDRESS TRANSACTIONS (P2SH-P2WPKH)

WORKFLOW

1. We generated three P2SH-SegWit addresses: A', B', and C'
2. Funded address A' using `sendtoaddress`
3. Created a raw transaction from A' to B' using `createrawtransaction`
4. Decoded the transaction to examine the locking script for B'
5. Signed and broadcast the transaction, generating txid: <txid_a'_to_b'>
6. Retrieved the unspent transaction output (UTXO) for address B'
7. Created a transaction from B' to C' using this UTXO
8. Decoded, signed, and broadcast this transaction, generating txid: <txid_b'_to_c'>

SCRIPT ANALYSIS

Transaction A' to B':

The ScriptPubKey (locking script) for address B' follows the P2SH-P2WPKH pattern:

```
OP_HASH160 <script hash> OP_EQUAL
```

This script does the following:

1. Hashes the provided redeem script
2. Checks if it matches the stored script hash

```
PS C:\Program Files\Bitcoin\daemon> .\bitcoin-cli.exe -regtest decoderawtransaction (.\bitcoin-cli.exe -regtest getrawtransaction "15f00e0da6d82c54054ea3939
fce946338d03ac306fe87172271759b00871c30")
{
  "txid": "15f00e0da6d82c54054ea3939fce946338d03ac306fe87172271759b00871c30",
  "hash": "4a2de6d05957d71d0f3bb6d060dd652081f25365fd4904df5a8c8c33925e5f13",
  "version": 2,
  "size": 215,
  "vsize": 134,
  "weight": 533,
  "locktime": 0,
  "vin": [
    {
      "txid": "0e3bec0bf71a57a2d98ef43cc09d4daa14c73a21ca79580a9871ae5e6a189044",
      "vout": 1,
      "scriptSig": {
        "asm": "0014357d7f6e5c965afbbaaf117d4beb87b429c128dc",
        "hex": "160014357d7f6e5c965afbbaaf117d4beb87b429c128dc"
      },
      "txinwitness": [
        "304402207d0a18a97bee7cba943ef31f1f0eda2ccb6bf11f430325f77ca89ad3cc39f8bf02206a3b397937d030dc1323d0c6dc555164de3f84a3d7ce9d4ee97f6525f4df60f201",
        "032763010df85cd11d7e73240a49cce16a455776463e71490cb740bb0f9e993a48"
      ],
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": 0.99990000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_HASH160 3823cd961dce36366cbcdc63ace32b1fe5bb0ec0 OP_EQUAL",
        "desc": "addr(2MxN4iffihUoJ8WkqgnDCHpdTu6vrGnzeks)#x23gvyn5",
        "hex": "a9143823cd961dce36366cbcdc63ace32b1fe5bb0ec087",
        "address": "2MxN4iffihUoJ8WkqgnDCHpdTu6vrGnzeks",
        "type": "scripthash"
      }
    }
  ]
}
```

Transaction from A' to B'

Transaction B' to C':

For P2SH-SegWit, the ScriptSig (unlocking script) contains only the redeem script:

```
<redeem script>
```

The witness data (not part of the ScriptSig) contains:

```
<signature> <public key>
```

The redeem script is typically:

```
OP_0 OP_HASH160 <pubkey hash> OP_EQUAL
```

where the witness program is the HASH160 of the public key.

```
PS C:\Program Files\Bitcoin\daemon> .\bitcoin-cli.exe -regtest decoderawtransaction (.\bitcoin-cli.exe -regtest getrawtransaction "229ed081cf9af34e051220180
84139b7e05d40ae4b3893357e99ca6c555e5916")
{
  "txid": "229ed081cf9af34e05122018084139b7e05d40ae4b3893357e99ca6c555e5916",
  "hash": "c7eba849af038cfb540083137eef2bb787e0253cebeec9e965f77975b29b6eeb",
  "version": 2,
  "size": 215,
  "vsize": 134,
  "weight": 533,
  "locktime": 0,
  "vin": [
    {
      "txid": "15f00e0da6d82c54054ea3939fce946338d03ac306fe87172271759b00871c30",
      "vout": 0,
      "scriptSig": {
        "asm": "0014263e7f5fbb9155682a1af5621c00937d01b4bc5d",
        "hex": "160014263e7f5fbb9155682a1af5621c00937d01b4bc5d"
      },
      "txinwitness": [
        "304402202696bb1f262c4ff1edd5b3c55aac47c4f6220c85f8c9d1650e94ba8da7c46fbb022063b39026e8e778f8766a557d3c134d5d5cd046060617be4129d86d2df52d34cf01",
        "0213b23d454cde67be72e71751ff76d9b6d67225a6894857eb52e7357816d4eb77"
      ],
      "sequence": 4294967293
    }
  ],
  "vout": [
    {
      "value": 0.99980000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_HASH160 908800948edc0e01f56b5577725cc6f24eda4731 OP_EQUAL",
        "desc": "addr(2N6RSDWaE9FmzrYAn8rA2mzK4PHuAapS3LD)#z82qy42p",
        "hex": "a914908800948edc0e01f56b5577725cc6f24eda473187",
        "address": "2N6RSDWaE9FmzrYAn8rA2mzK4PHuAapS3LD",
        "type": "scripthash"
      }
    }
  ]
}
```

Transaction from B’ to C’

When executed with the locking script from the previous transaction, it forms a complete validation script:

`<redeem script> OP_HASH160 <script hash> OP_EQUAL`

Challenge-Response Validation using `btcdeb`:

```
cse@cse-HP-ProDesk-600-G5-PCI-MT:~/Desktop/btcdeb/btcdeb$ btcdeb --verbose '[0014263e7f5fbb9155682a1af5621c00937d01b4bc5d OP_HASH160 3823cd961dce36366cbcdc63ace32b1fe5bb0ec0 OP_EQUAL]'
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
-------------------------------------------+--------
0014263e7f5fbb9155682a1af5621c00937d01b4bc5d |
OP_HASH160                                 |
3823cd961dce36366cbcdc63ace32b1fe5bb0ec0   |
OP_EQUAL                                   |
#0000 0014263e7f5fbb9155682a1af5621c00937d01b4bc5d
```

Loading the Script

1. Push redeem script onto the stack

```
btcdeb> step
        <> PUSH stack 0014263e7f5fbb9155682a1af5621c00937d01b4bc5d
script                                     |                              stack
-------------------------------------------+-------------------------------------------
OP_HASH160                                 | 0014263e7f5fbb9155682a1af5621c00937d01b4bc5d
3823cd961dce36366cbcdc63ace32b1fe5bb0ec0   |
OP_EQUAL                                   |
#0001 OP_HASH160
```

2. OP_HASH160 hashes the redeem script

```
btcdeb> step
        <> POP  stack
        <> PUSH stack 3823cd961dce36366cbcdc63ace32b1fe5bb0ec0
script                                     |                              stack
-------------------------------------------+-------------------------------------------
3823cd961dce36366cbcdc63ace32b1fe5bb0ec0   |     3823cd961dce36366cbcdc63ace32b1fe5bb0ec0
OP_EQUAL                                   |
#0002 3823cd961dce36366cbcdc63ace32b1fe5bb0ec0
```

3. Push script hash onto the stack

```
btcdeb> step
        <> PUSH stack 3823cd961dce36366cbcdc63ace32b1fe5bb0ec0
script                                     |                              stack
-------------------------------------------+-------------------------------------------
OP_EQUAL                                   |     3823cd961dce36366cbcdc63ace32b1fe5bb0ec0
                                           |     3823cd961dce36366cbcdc63ace32b1fe5bb0ec0
#0003 OP_EQUAL
```

4. OP_EQUAL compares the hash of the redeem script with the stored script hash

```
btcdeb> step
        <> POP  stack
        <> POP  stack
        <> PUSH stack 01
script                                     |                              stack
-------------------------------------------+-------------------------------------------
                                           |                                         01
```

5. The top of the stack gets 01 if the result of OP_EQUAL evaluates to true.

```
btcdeb> step
script                                     |                              stack
-------------------------------------------+-------------------------------------------
                                           |                                         01
btcdeb> step
at end of script
```

PART 3: COMPARISON AND ANALYSIS

Transaction Type	Size (bytes)	Virtual Size (vbytes)	Weight Units
Legacy A to B	191	191	764
Legacy B to C	191	191	764
SegWit A' to B'	215	134	533
SegWit B' to C'	215	134	533

The virtual size reduction is 29.84%

SCRIPT STRUCTURE DIFFERENCES

1. *P2PKH (Legacy)*
    - ScriptPubKey

    `OP_DUP OP_HASH160 <pubkey hash> OP_EQUALVERIFY OP_CHECKSIG`
    - ScriptSig

    `<signature> <public key>`

    All validation happens in a single script execution

2. *P2SH-P2WPKH (SegWit)*
    - ScriptPubKey

    `OP_HASH160 [script hash] OP_EQUAL`
    - ScriptSig

    `<redeem script>`
    - Witness Data

    `<signature> <public key>`
    - Validation happens in multiple steps:
        1. Verify redeem script hash
        2. Execute redeem script
        3. Verify witness data separately

BENEFITS OF SEGWIT

1. *Transaction Malleability Fix:* By moving signatures to witness data, the transaction ID is no longer affected by signature modifications and hence second or third party attacks.
2. *Scalability:* SegWit effectively increases the block capacity without changing the block size limit. The smaller virtual sizes result in lower fees despite similar or larger raw byte sizes.
3. *Fee Efficiency:* Witness data is discounted in fee calculations, incentivizing the use of SegWit.

GitHub Repository Link: https://github.com/Srinidhi-Sai-Boorgu/MineOverMatter-Bitcoin-Scripting

---

END