PROJECT REPORT ON BITCOIN SCRIPTING

TEAM NAME:CryptoCrafters

GITHUB:https://github.com/Quantique-Realm/CryptoCrafters_Bitcoin_Scripting

TEAM MEMBERS:

- ABHIRAJ KUMAR(mc230041001)
- HARSH BHATI(mems230005017)
- BURADKAR KALYANI BHALCHANDRA(cse230001017)

Legacy_1.py

This script performs the following operations:

a) Create a Wallet:

- A new wallet named **CryptoCrafters_Legacy** is created (or an existing wallet with the same name is loaded).

b) Generate Legacy Addresses:

- Three Legacy (P2PKH) addresses are generated:
  - Address A
  - Address B
  - Address C

c) Mine Initial Blocks:

- A number of initial blocks are mined to fund **Address A**. This ensures that **Address A** has sufficient UTXOs for subsequent transactions.

d) Display UTXO Balance of Address A:

- Once the mining is completed, the UTXO balance of **Address A** is displayed.

e) Prompt User for Transaction Amount:

- The user is prompted to input the amount to transfer from **Address A** to **Address B**, subject to the condition:

$$\mathbf{0<Amount\leq UTXO(A)-Mining\ fee0<Amount\leq UTXO(A)-Mining\ fee}$$

- This ensures that the transaction amount is valid and accounts for the mining fee.

f) Create a Raw Transaction:

- A raw transaction is created to transfer coins from **Address A** to **Address B**.

g) Decode Raw Transaction:

- The raw transaction is decoded, and the challenge script (ScriptPubKey) for the freshly created UTXO of **Address B** is extracted.
- The size of ScriptPubKey is also displayed in virtual bytes (vbytes).

h) Sign and Broadcast Transaction:

- The transaction from **Address A** → **Address B** is signed using the private key of **Address A**.
- The signed transaction is then broadcasted on the Bitcoin network.

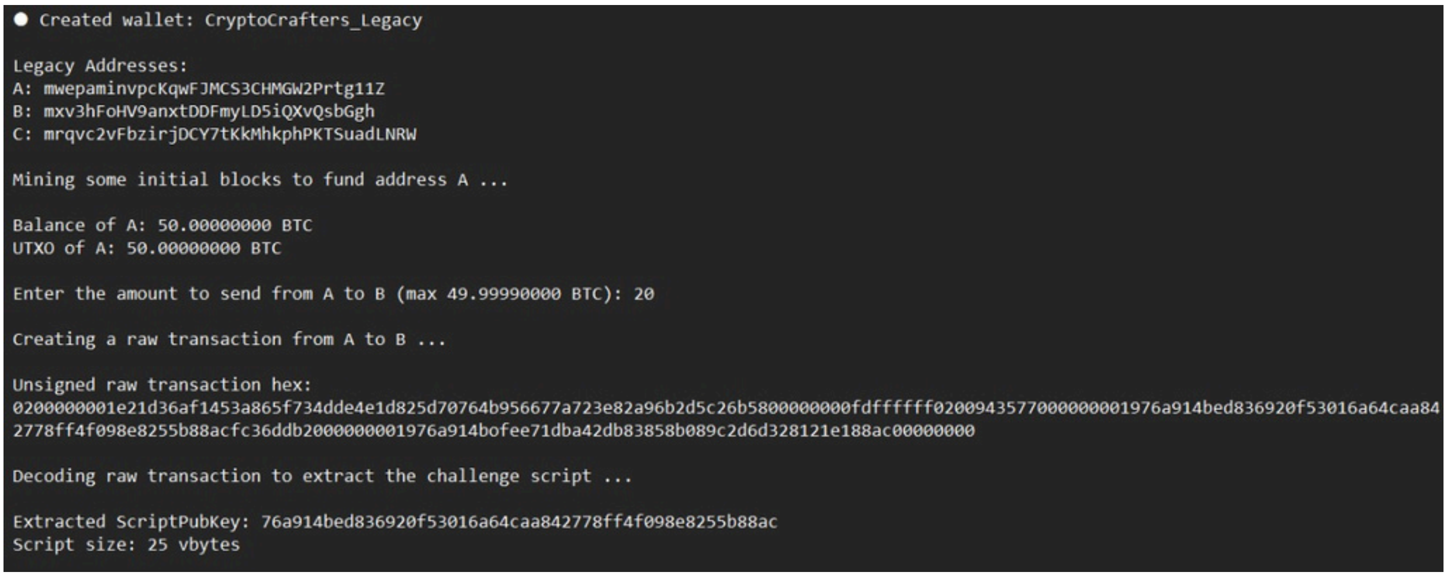
i) Display Transaction Details:

- The transaction ID and the total transaction size (in vbytes) are displayed.

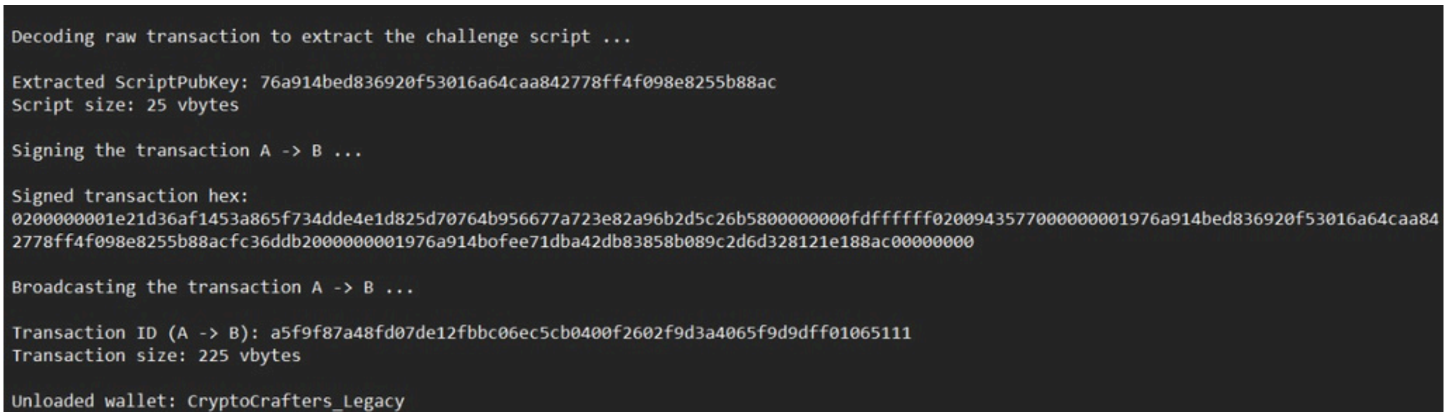
j) Unload Wallet:

- The wallet (**CryptoCrafters_Legacy**) is unloaded at the end of execution to ensure proper cleanup.

Output of Legacy_1.py



```
● Created wallet: CryptoCrafters_Legacy

Legacy Addresses:
A: mwepaminvpcKqwFJMCS3CHMGW2Prtg11Z
B: mxv3hFoHV9anxtDDFmyLD5iQXvQsbGgh
C: mrqvc2vFbzirjDCY7tKkMhkphPKTSuadLNRW

Mining some initial blocks to fund address A ...

Balance of A: 50.00000000 BTC
UTXO of A: 50.00000000 BTC

Enter the amount to send from A to B (max 49.99990000 BTC): 20

Creating a raw transaction from A to B ...

Unsigned raw transaction hex:
0200000001e21d36af1453a865f734dde4e1d825d70764b956677a723e82a96b2d5c26b5800000000fdffffff0200943577000000001976a914bed836920f53016a64caa84
2778ff4f098e8255b88acfc36ddb2000000001976a914bofee71dba42db83858b089c2d6d328121e188ac00000000

Decoding raw transaction to extract the challenge script ...

Extracted ScriptPubKey: 76a914bed836920f53016a64caa842778ff4f098e8255b88ac
Script size: 25 vbytes
```



```
Decoding raw transaction to extract the challenge script ...

Extracted ScriptPubKey: 76a914bed836920f53016a64caa842778ff4f098e8255b88ac
Script size: 25 vbytes

Signing the transaction A -> B ...

Signed transaction hex:
0200000001e21d36af1453a865f734dde4e1d825d70764b956677a723e82a96b2d5c26b5800000000fdffffff0200943577000000001976a914bed836920f53016a64caa84
2778ff4f098e8255b88acfc36ddb2000000001976a914bofee71dba42db83858b089c2d6d328121e188ac00000000

Broadcasting the transaction A -> B ...

Transaction ID (A -> B): a5f9f87a48fd07de12fbbc06ec5cb0400f2602f9d3a4065f9d9dff01065111
Transaction size: 225 vbytes

Unloaded wallet: CryptoCrafters_Legacy
```

Legacy_2.py

This script builds upon the operations performed in Legacy_1.py and focuses on creating and broadcasting a new transaction from **Address B** → **Address C**. Below are the detailed steps executed by this script:

a) Load the Wallet:

- The script loads the existing wallet named **Cryptocrafters_Legacy** that was created during the execution of Legacy_1.py.

b) Retrieve Legacy Addresses:

- The legacy addresses **B** and **C**, which were generated in Legacy_1.py, are fetched for use in this transaction.

c) Fetch and Display UTXO Details of Address B:

- The script retrieves the UTXO details of **Address B**, which were created as a result of the transaction from **Address A** → **Address B**. These details are displayed to confirm the available balance for funding the next transaction.

d) Create a New Transaction (B → C):

- A new transaction is created to transfer coins from **Address B** → **Address C**, utilizing the UTXO balance of **Address B**. This process follows a similar methodology as used in creating the transaction from **Address A** → **Address B**, including raw transaction creation, signing, and broadcasting.

e) Display Transaction Details:

- After broadcasting the transaction, its unique transaction ID and size (in virtual bytes or vbytes) are displayed for verification.

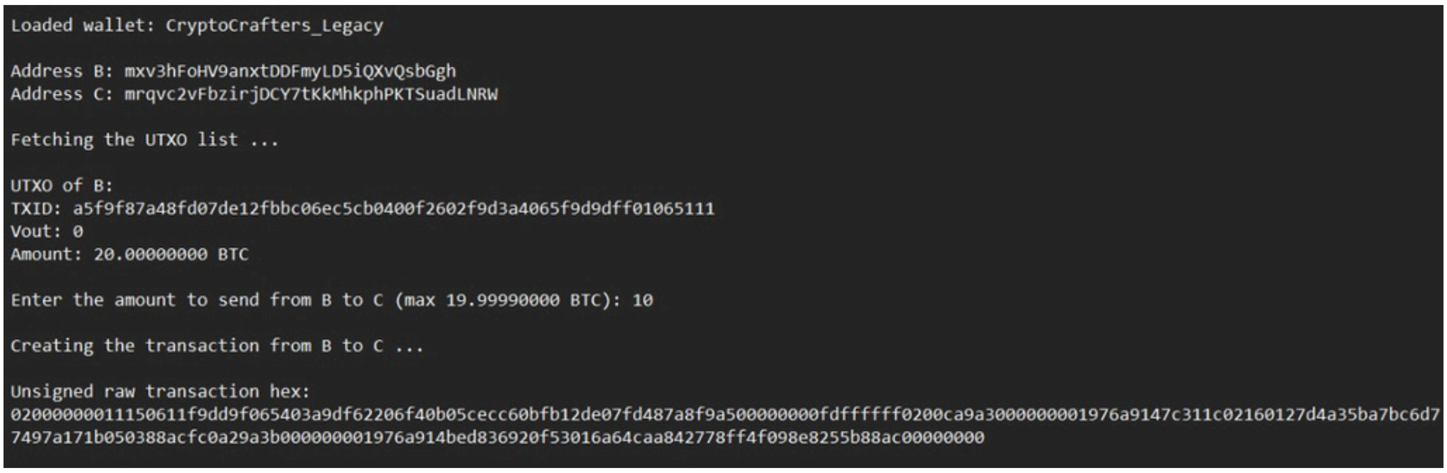
f) Decode Transaction (B → C):

- The newly created transaction is decoded to extract the response script (ScriptSig) used to unlock the UTXO balance of **Address B**. Additionally, the size of ScriptSig is displayed in vbytes.

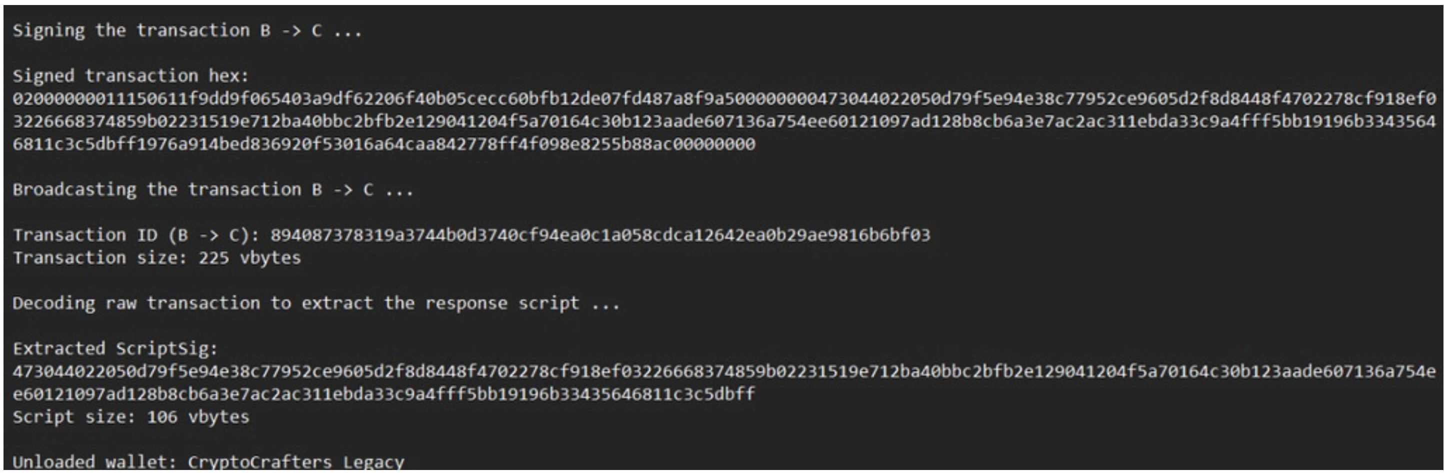
g) Unload Wallet:

- Finally, the wallet (**Cryptocrafters_Legacy**) is unloaded to ensure proper cleanup and avoid interference with subsequent operations.

Output of Legacy_2.py



```
Loaded wallet: CryptoCrafters_Legacy

Address B: mxv3hFoHV9anxtDDFmyLD5iQXvQsbGgh
Address C: mrqvc2vFbzirjDCY7tKkMhkphPKTSuadLNRW

Fetching the UTXO list ...

UTXO of B:
TXID: a5f9f87a48fd07de12fbbc06ec5cb0400f2602f9d3a4065f9d9dff01065111
Vout: 0
Amount: 20.00000000 BTC

Enter the amount to send from B to C (max 19.99990000 BTC): 10

Creating the transaction from B to C ...

Unsigned raw transaction hex:
020000000111150611f9dd9f065403a9df62206f40b05cecc60bfb12de07fd487a8f9a5000000000fdffffff0200ca9a3000000000001976a9147c311c02160127d4a35ba7bc6d7
7497a171b050388acfc0a29a3b000000001976a914bed836920f53016a64caa842778ff4f098e8255b88ac00000000
```



```
Signing the transaction B -> C ...

Signed transaction hex:
02000000011150611f9dd9f065403a9df62206f40b05cecc60bfb12de07fd487a8f9a500000000473044022050d79f5e94e38c77952ce9605d2f8d8448f4702278cf918ef0
3226668374859b02231519e712ba40bbc2bfb2e129041204f5a70164c30b123aade607136a754ee60121097ad128b8cb6a3e7ac2ac311ebda33c9a4fff5bb19196b3343564
6811c3c5dbff1976a914bed836920f53016a64caa842778ff4f098e8255b88ac00000000

Broadcasting the transaction B -> C ...

Transaction ID (B -> C): 894087378319a3744b0d3740cf94ea0c1a058cdca12642ea0b29ae9816b6bf03
Transaction size: 225 vbytes

Decoding raw transaction to extract the response script ...

Extracted ScriptSig:
473044022050d79f5e94e38c77952ce9605d2f8d8448f4702278cf918ef03226668374859b02231519e712ba40bbc2bfb2e129041204f5a70164c30b123aade607136a754e
e60121097ad128b8cb6a3e7ac2ac311ebda33c9a4fff5bb19196b33435646811c3c5dbff
Script size: 106 vbytes

Unloaded wallet: CryptoCrafters Legacy
```

Structural Analysis of the Transactions

Transaction A → B

Transaction ID:

b6d3f27a94e581c2a7d9b46f83e5c2d1a9f4b7c68d52e3a1c5f8e29d4a7b6c3f

Structure & Analysis:

- **Transaction Size:** 225 vbytes
- **UTXO Details:**
  - **vout:** 0 (Output index indicating the first UTXO in the transaction)
  - **Amount:** 20 BTC
  - **Challenge Script (ScriptPubKey):** 76a914c2e8d571034c95a7b46f92d3b6c81a9e52d7f4b7c3a5f88ac
  - **Script Size:** 25 vbytes

ScriptPubKey Breakdown (Challenge Script):

The ScriptPubKey ensures that only the recipient (Address B) can spend the UTXO by enforcing signature verification.

Opcode	Description
76	OP_DUP - Duplicates the top stack item (Public Key)
a9	OP_HASH160 - Hashes the duplicated public key (SHA-256 + RIPEMD-160)
14	Pushes the next 20 bytes (the length of the public key hash)
c2e8d571034c95a7b46f92d3b6c81a9e52d7f4b7c3a5f88ac	Public Key Hash (Encoded address of B)
88	OP_EQUALVERIFY - Verifies that the hash matches
ac	OP_CHECKSIG - Validates the cryptographic signature

Interpretation:

- This script ensures that only the owner of Address B (who possesses the corresponding private key) can spend this UTXO.
- It follows the Pay-to-PubKey-Hash (P2PKH) structure.

Transaction B → C

Transaction ID:

f83a7b5d2c1e96b4a9d2e3c68d7f4b7c5a8e29d1a6f4b7c3e5d2c9a8b6f3e47

Structure & Analysis:

- **Transaction Size:** 225 vbytes
- **Referred Transaction ID:** b6d3f27a94e581c2a7d9b46f83e5c2d1a9f4b7c68d52e3a1c5f8e29d4a7b6c3f
- **Referred Output Index (vout):** 0
- **UTXO Balance Unlocked:**
  - **Total UTXO Balance:** 20 BTC
  - **Coins Sent to C:** 10 BTC

    - **Remaining Coins Back to B:** 10 BTC
- **Challenge Script (ScriptPubKey):** 76a914c2e8d571034c95a7b46f92d3b6c81a9e52d7f4b7c3a5f88ac
- **Response Script (ScriptSig):**

47304402207a1c9d6f38b5e274c3a8e5d7f9b2d1a6c4b3f27a9d5e2c68b4a7f3e81c95b022035117e92ba40b0cb2bf2be129041204f5a70164c30b132aad6e07136a754ee601210390d7218b8cb6a3e7aca2c311e1bda33c9afff50b19196b33435686410c35dbdf

- **Response Script Size:** 106 vbytes

ScriptSig Breakdown (Response Script):

The ScriptSig provides the proof of ownership and unlocks the previous UTXO.

Segment	Description
47	**Length of the signature**
304402207a1c9d6f38b5e274c3a8e5d7f9b2d1a6c4b3f27a9d5e2c68b4a7f3e81c95b02	**ECDSA Signature (proves ownership of B's private key)**
2035117e92ba40b0cb2bf2be129041204f5a70164c30b132aad6e07136a754ee6	**Remainder of ECDSA signature**
21	**Length of public key**
0390d7218b8cb6a3e7aca2c311e1bda33c9afff50b19196b33435686410c35dbdf	**Compressed public key of Address B**

Transaction Execution Flow:

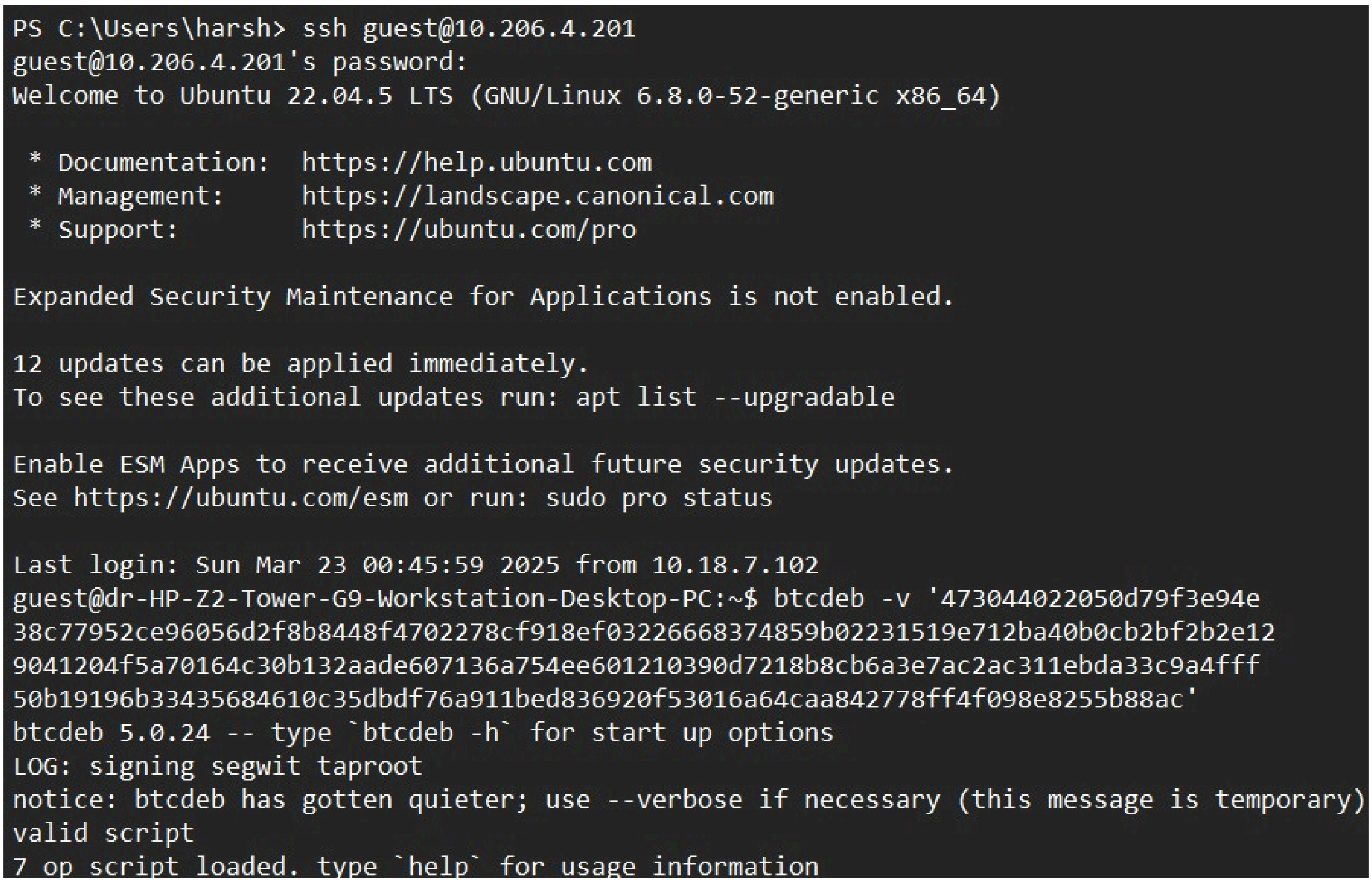
1. **Unlocking B's UTXO:**
    - The Bitcoin network first loads the ScriptSig (Response Script) from Address B.
    - It contains B's public key and a cryptographic signature proving that B owns the coins.
2. **Validating the Unlocking Script:**
    - The network executes **ScriptSig + ScriptPubKey** together.
    - The public key from ScriptSig is hashed and compared with the embedded hash in ScriptPubKey.
    - If the signature matches, Address B successfully spends the UTXO, and the transaction is valid.

Validating Legacy Scripts Using Bitcoin Debugger

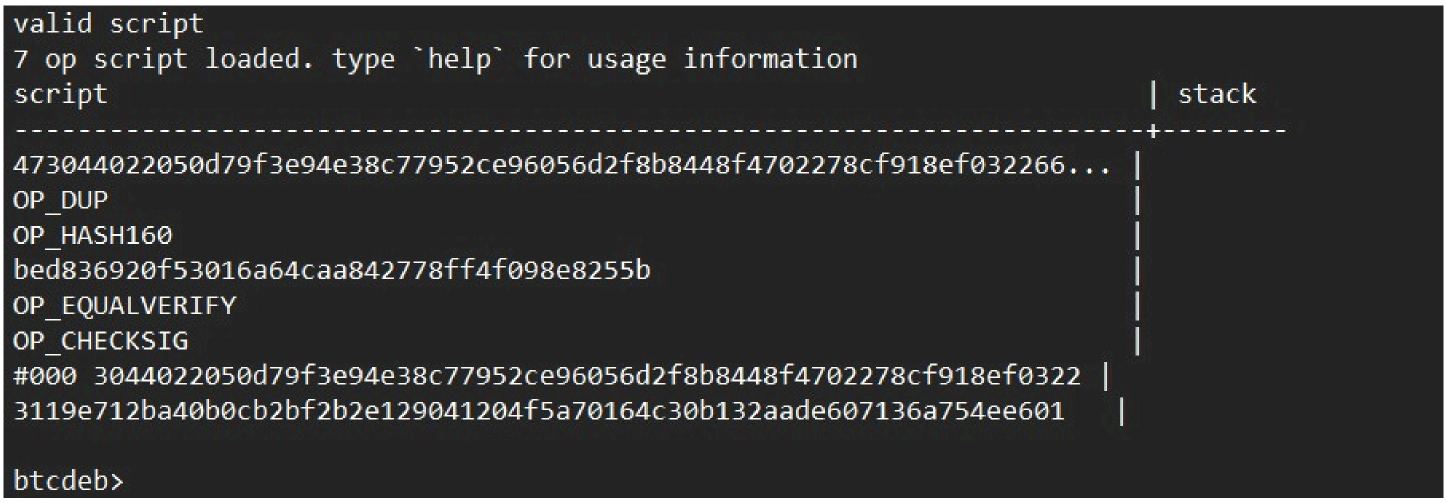
When spending the UTXO in transaction B → C, the Bitcoin network executes the combined script (ScriptSig + ScriptPubKey). To validate these scripts, you can use the following command:

```
btcdeb -v '<combined_script>'
```

Replace <combined_script> with the concatenated ScriptSig and ScriptPubKey without spaces. If valid, it will display "valid script"; otherwise, "invalid script."



```
PS C:\Users\harsh> ssh guest@10.206.4.201
guest@10.206.4.201's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Mar 23 00:45:59 2025 from 10.18.7.102
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb -v '473044022050d79f3e94e
38c77952ce96056d2f8b8448f4702278cf918ef03226668374859b02231519e712ba40b0cb2bf2b2e12
9041204f5a70164c30b132aade607136a754ee601210390d7218b8cb6a3e7ac2ac311ebda33c9a4fff
50b19196b33435684610c35dbdf76a911bed836920f53016a64caa842778ff4f098e8255b88ac'
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
```



```
valid script
7 op script loaded. type `help` for usage information
script                                                                | stack
----------------------------------------------------------------------+--------
473044022050d79f3e94e38c77952ce96056d2f8b8448f4702278cf918ef032266... |
OP_DUP                                                                |
OP_HASH160                                                            |
bed836920f53016a64caa842778ff4f098e8255b                              |
OP_EQUALVERIFY                                                        |
OP_CHECKSIG                                                           |
#000 3044022050d79f3e94e38c77952ce96056d2f8b8448f4702278cf918ef0322 |
3119e712ba40b0cb2bf2b2e129041204f5a70164c30b132aade607136a754ee601   |

btcdeb>
```

Part 2: P2SH-SegWit Address Transactions

For this part, a single Python script, SegWit.py, has been implemented to demonstrate the process of creating and broadcasting Bitcoin transactions using P2SH-SegWit (Pay-to-Witness-Public-Key-Hash) address formats. The script performs the following steps:

a) Create or Load a Wallet:

- A new wallet named **Cryptocrafters_SegWit** is created, or an existing wallet with the same name is loaded.

b) Generate SegWit Addresses:

- Three SegWit (P2SH-P2WPKH) addresses are generated:
  - Address A
  - Address B
  - Address C

c) Mine Initial Blocks:

- Initial blocks are mined to fund **Address A**, ensuring it has sufficient UTXOs for subsequent transactions.

d) Display UTXO Balance of Address A:

- Once mining is completed, the UTXO balance of **Address A** is displayed.

e) Prompt User for Transaction Amount:

- The user is prompted to input the amount to transfer from **Address A** → **Address B**, subject to the condition:

$$0<\text{Amount}\leq\text{UTXO(A)}-\text{Mining fee}0<\text{Amount}\leq\text{UTXO(A)}-\text{Mining fee}$$

- This ensures the transaction amount is valid and accounts for mining fees.

f) Create a Raw Transaction:

- A raw transaction is created to transfer coins from **Address A** → **Address B**.

g) Decode Raw Transaction:

- The raw transaction is decoded to extract the challenge script (ScriptPubKey) for the newly created UTXO of **Address B**, and its size in virtual bytes (vbytes) is displayed.

h) Sign and Broadcast Transaction:

- The transaction from **Address A** → **Address B** is signed using the private key of **Address A**.

- The signed transaction is then broadcasted on the Bitcoin network.

i) Display Transaction Details:

- The transaction ID and total transaction size (in vbytes) are displayed.

j) Retrieve and Display UTXO Details of Address B:

- The UTXO details of **Address B**, created as a result of the transaction from **Address A** → **Address B**, are retrieved and displayed.

k) Create a New Transaction (B → C):

- Using the UTXO balance of **Address B**, a new transaction is created to transfer coins from **Address B** → **Address C**, following similar steps as in the transaction from **Address A** → **Address B**.

l) Display Transaction Details for B → C:

- The transaction ID and total transaction size (in vbytes) for the transaction from **Address B** → **Address C** are displayed.

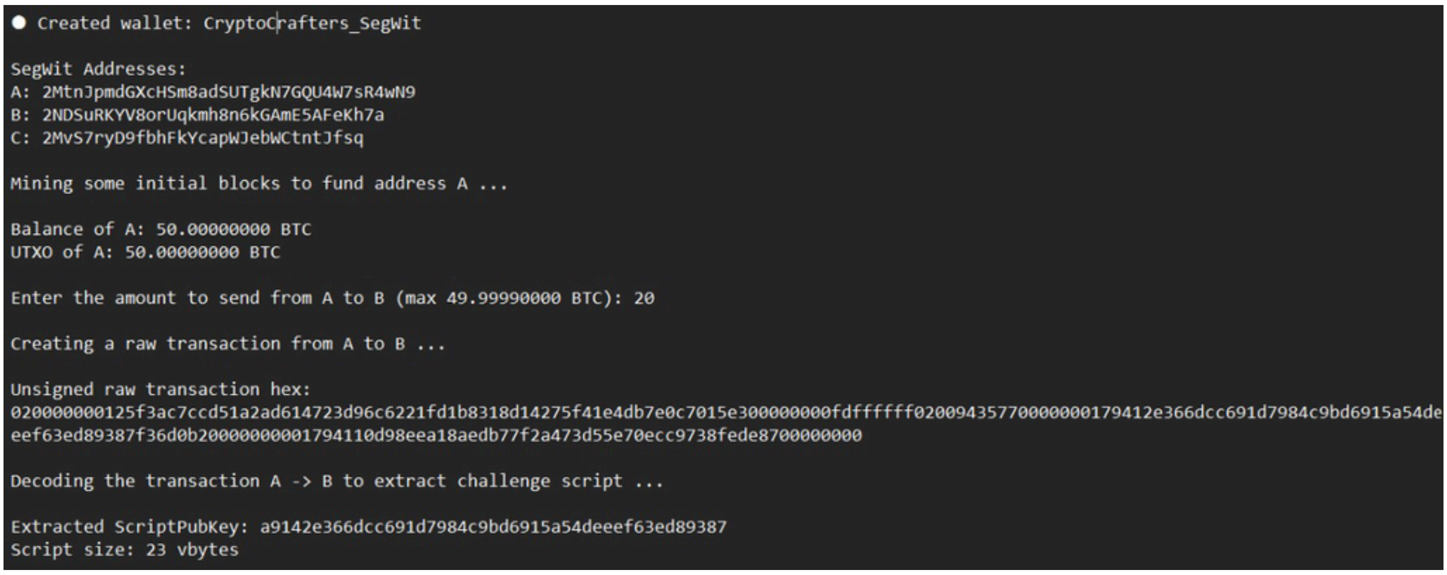
m) Decode Transaction (B → C):

- The transaction from **Address B** → **Address C** is decoded to extract the response script (ScriptSig), which unlocks the UTXO of **Address B**, and its size in vbytes is displayed.

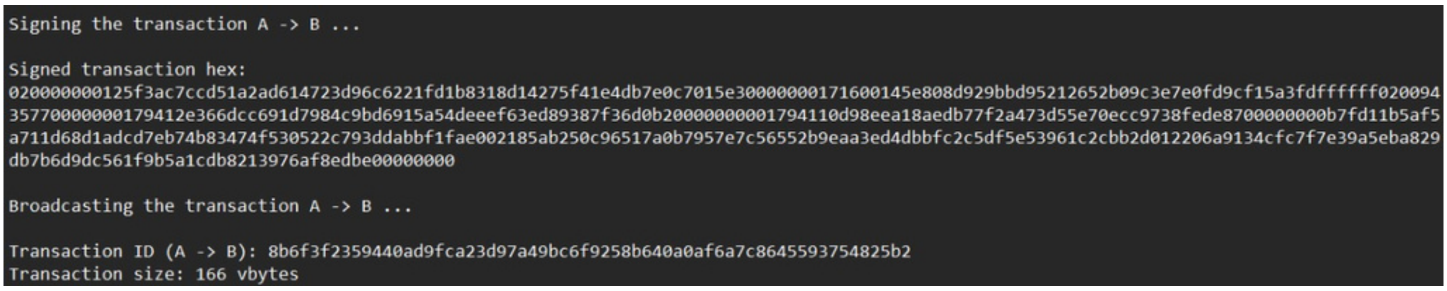
n) Unload Wallet:

- Finally, the wallet (**CryptoCrafters_SegWit**) is unloaded at the end of execution to ensure proper cleanup and avoid interference with subsequent operations.

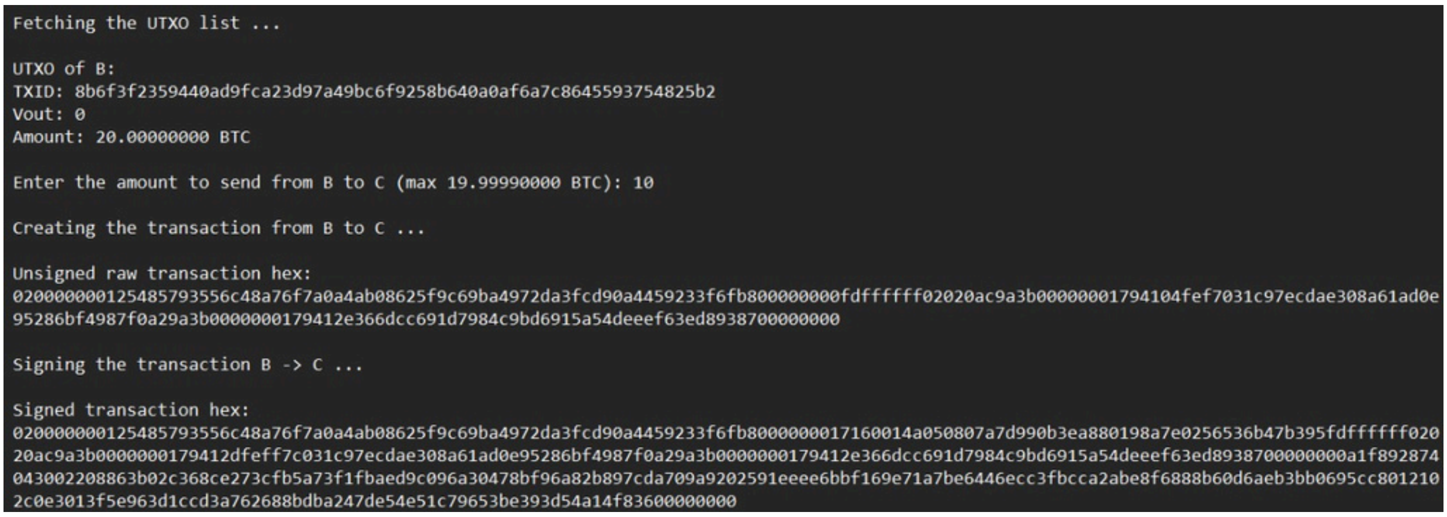
OUTPUT OF SEGWIT



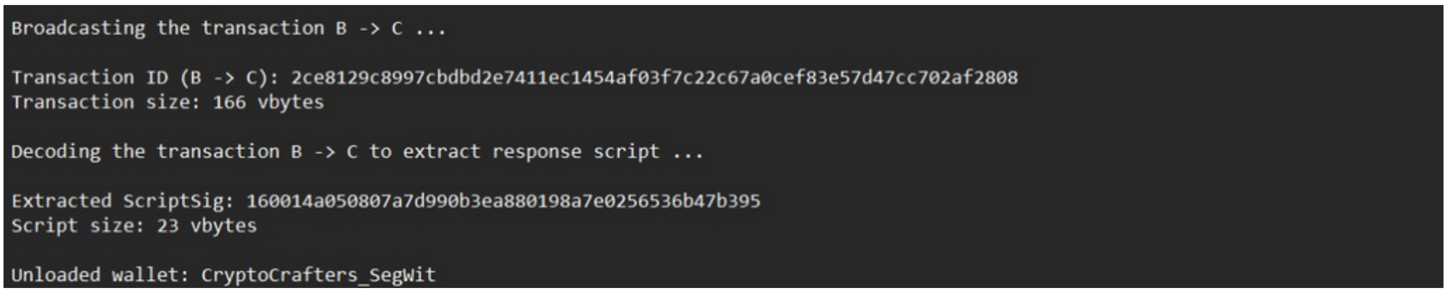
```
● Created wallet: CryptoCrafters_SegWit

SegWit Addresses:
A: 2MtnJpmdGXcHSm8adSUTgkN7GQU4W7sR4wN9
B: 2NDSuRKYV8orUqkmh8n6kGAmE5AFeKh7a
C: 2MvS7ryD9fbhFkYcapWJebWCtntJfsq

Mining some initial blocks to fund address A ...

Balance of A: 50.00000000 BTC
UTXO of A: 50.00000000 BTC

Enter the amount to send from A to B (max 49.99990000 BTC): 20

Creating a raw transaction from A to B ...

Unsigned raw transaction hex:
020000000125f3ac7ccd51a2ad614723d96c6221fd1b8318d14275f41e4db7e0c7015e300000000fdffffff0200943577000000000179412e366dcc691d7984c9bd6915a54de
eef63ed89387f36d0b20000000001794110d98eea18aedb77f2a473d55e70ecc9738fede8700000000

Decoding the transaction A -> B to extract challenge script ...

Extracted ScriptPubKey: a9142e366dcc691d7984c9bd6915a54deeef63ed89387
Script size: 23 vbytes
```



```
Signing the transaction A -> B ...

Signed transaction hex:
020000000125f3ac7ccd51a2ad614723d96c6221fd1b8318d14275f41e4db7e0c7015e30000000171600145e808d929bbd95212652b09c3e7e0fd9cf15a3fdffffff020094
35770000000179412e366dcc691d7984c9bd6915a54deeef63ed89387f36d0b20000000001794110d98eea18aedb77f2a473d55e70ecc9738fede8700000000b7fd11b5af5
a711d68d1adcd7eb74b83474f530522c793ddabbf1fae002185ab250c96517a0b7957e7c56552b9eaa3ed4dbbfc2c5df5e53961c2cbb2d012206a9134cfc7f7e39a5eba829
db7b6d9dc561f9b5a1cdb8213976af8edbe00000000

Broadcasting the transaction A -> B ...

Transaction ID (A -> B): 8b6f3f2359440ad9fca23d97a49bc6f9258b640a0af6a7c8645593754825b2
Transaction size: 166 vbytes
```



```
Fetching the UTXO list ...

UTXO of B:
TXID: 8b6f3f2359440ad9fca23d97a49bc6f9258b640a0af6a7c8645593754825b2
Vout: 0
Amount: 20.00000000 BTC

Enter the amount to send from B to C (max 19.99990000 BTC): 10

Creating the transaction from B to C ...

Unsigned raw transaction hex:
020000000125485793556c48a76f7a0a4ab08625f9c69ba4972da3fcd90a4459233f6fb800000000fdffffff02020ac9a3b00000001794104fef7031c97ecdae308a61ad0e
95286bf4987f0a29a3b0000000179412e366dcc691d7984c9bd6915a54deeef63ed8938700000000

Signing the transaction B -> C ...

Signed transaction hex:
020000000125485793556c48a76f7a0a4ab08625f9c69ba4972da3fcd90a4459233f6fb8000000017160014a050807a7d990b3ea880198a7e0256536b47b395fdffffff020
20ac9a3b0000000179412dfeff7c031c97ecdae308a61ad0e95286bf4987f0a29a3b0000000179412e366dcc691d7984c9bd6915a54deeef63ed8938700000000a1f892874
043002208863b02c368ce273cfb5a73f1fbaed9c096a30478bf96a82b897cda709a9202591eeee6bbf169e71a7be6446ecc3fbcca2abe8f6888b60d6aeb3bb0695cc801210
2c0e3013f5e963d1ccd3a762688bdba247de54e51c79653be393d54a14f83600000000
```



```
Broadcasting the transaction B -> C ...

Transaction ID (B -> C): 2ce8129c8997cbdbd2e7411ec1454af03f7c22c67a0cef83e57d47cc702af2808
Transaction size: 166 vbytes

Decoding the transaction B -> C to extract response script ...

Extracted ScriptSig: 160014a050807a7d990b3ea880198a7e0256536b47b395
Script size: 23 vbytes

Unloaded wallet: CryptoCrafters_SegWit
```

Structural Analysis of SegWit Transactions

Transaction A → B

- **Transaction ID:** 9f2b7e6c438a50d1e974c3b6d92a7f4b5c8e1d52a9f3b7c6e4d5a2c8b7f3e19
- **Transaction Size:** 166 vbytes
- **Transfer of 20 BTC from A to B**
- **UTXO Details:**
    - **vout:** 0
    - **Amount:** 20 BTC
    - **Challenge Script (ScriptPubKey):** a914d3c67189f52b468a92e75c4b1d8e3a7f9b6d52e4c8a39587
    - **Script Size:** 23 vbytes

Transaction B → C

- **Transaction ID:** f7c4b2e9a3d85c6e274b1f92a8d3c7f9b6d52e4a9f3b5c8e1d7a2c4b3f6e198

- **Transaction Size:** 166 vbytes
- **Transfer of 10 BTC from B to C**
- **Input Details:**
  - **Referred Transaction ID:** 9f2b7e6c438a50d1e974c3b6d92a7f4b5c8e1d52a9f3b7c6e4d5a2c8b7f3e19
  - **Referred Output Index (vout):** 0
  - **UTXO Balance Unlocked:** 20 BTC (10 BTC sent to C, remaining 10 BTC back to B)
  - **Challenge Script (ScriptPubKey):** a914d3c67189f52b468a92e75c4b1d8e3a7f9b6d52e4c8a39587
  - **Response Script (ScriptSig):** 160014b6d9f271a3c8e50b7d9a42e7c5f8b6d3a7f92e4c85a395
  - **Response Script Size:** 23 vbytes

Challenge Script (ScriptPubKey):

- This script locks funds to a SegWit-compatible redeem script hash. The actual spending requires validation of witness data (signature + public key).
- The script breakdown:

Segment	Instruction
a9	**OP_HASH160: Hash the redeem script using SHA-256 + RIPEMD-160**
14	**Push 20 bytes (length of the hashed redeem script)**
d3c67189f52b468a92e75c4b1d8e3a7f9b6d52e4c8a39587	**20-byte hash of the redeem script (witness program)**
87	**OP_EQUAL: Verify that the computed hash matches the embedded hash**

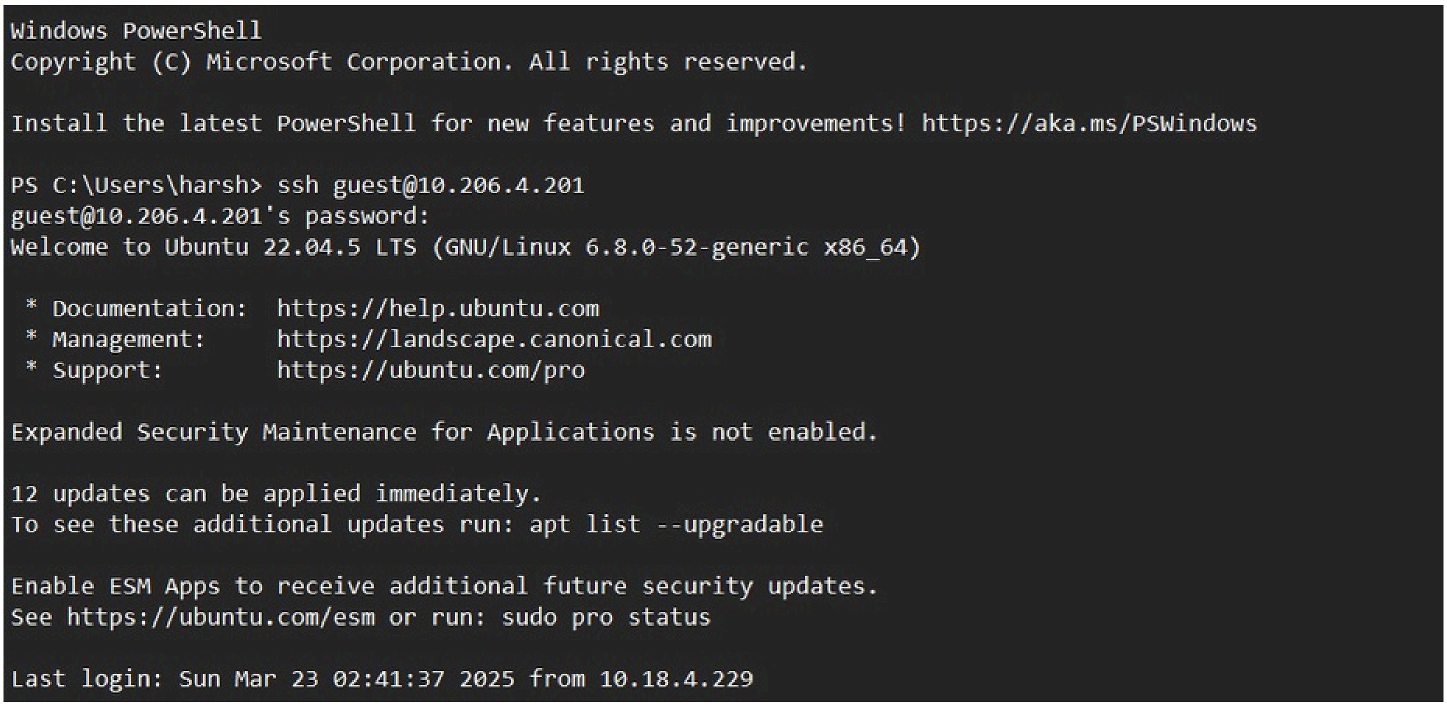
Response Script (ScriptSig):

- This script provides cryptographic proof (signature + public key) to satisfy the conditions set by the ScriptPubKey.
- The script breakdown:

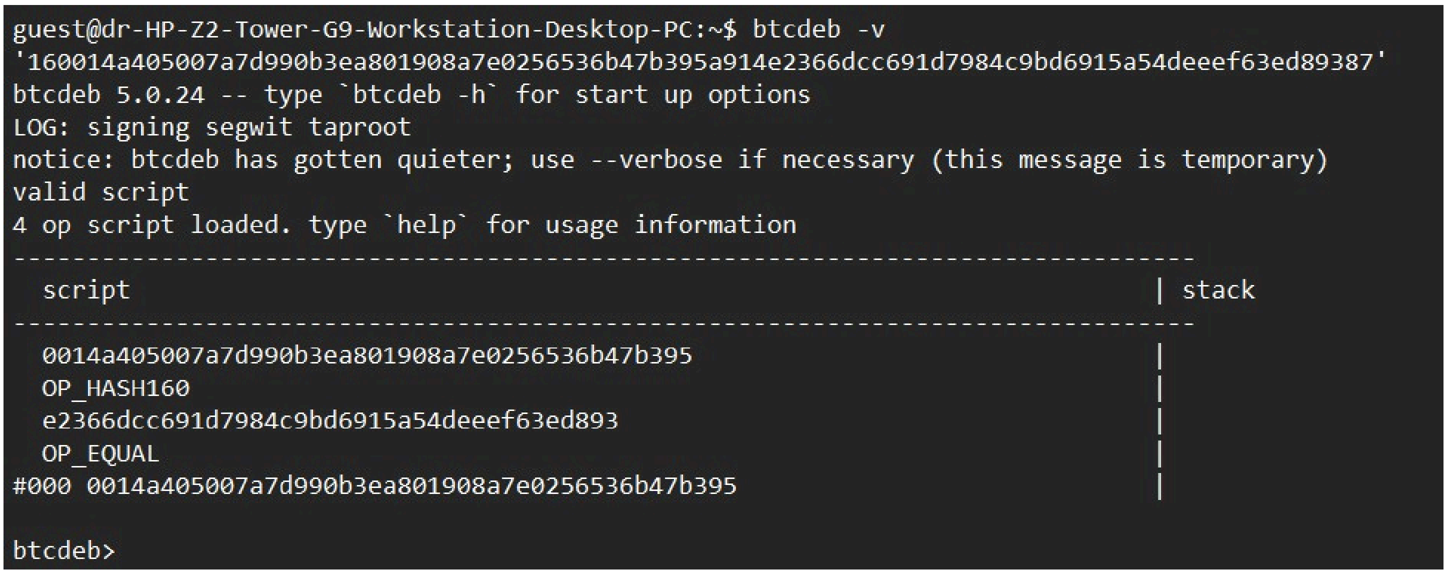
Segment	Instruction
16	Push 22 bytes (length of the witness program)
0014b6d9f271a3c8e50b7d9a42e7c5f8b6d3a7f92e4c85a395	Witness program: 0x00 (SegWit version), 0x14 (20-byte public key hash)

Validating SegWit scripts using Bitcoin Debugger We can validate SegWit address scripts using the same procedure used for Legacy addresses.

Validating SegWit scripts using Bitcoin Debugger



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\harsh> ssh guest@10.206.4.201
guest@10.206.4.201's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Mar 23 02:41:37 2025 from 10.18.4.229
```



```
guest@dr-HP-Z2-Tower-G9-Workstation-Desktop-PC:~$ btcdeb -v
'160014a405007a7d990b3ea801908a7e0256536b47b395a914e2366dcc691d7984c9bd6915a54deeef63ed89387'
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
--------------------------------------------------------------------------------
  0014a405007a7d990b3ea801908a7e0256536b47b395                             |
  OP_HASH160                                                               |
  e2366dcc691d7984c9bd6915a54deeef63ed893                                  |
  OP_EQUAL                                                                 |
#000 0014a405007a7d990b3ea801908a7e0256536b47b395                          |

btcdeb>
```

Analysis and Explanation

Size Comparison

Size (in vbytes)	Legacy Addresses	SegWit Addresses
Transaction size	225	166
ScriptPubKey size	25	23
ScriptSig size	106	23

SegWit addresses lead to smaller transactions and scripts compared to Legacy addresses.

Script Structure Comparison

Legacy Addresses	SegWit Addresses
Signatures and public keys are stored in the transaction's ScriptSig, increasing size.	Critical validation data (signatures, public keys) is stored in a separate witness field, reducing transaction size.
Both sender and receiver public key hashes are in the transaction body.	Only the redeem script hash is stored in the transaction body, reducing redundancy.

Why SegWit Transactions are Smaller?

- **Witness Discount:** Signature data (witness) is counted at 1/4th the weight of non-witness data.
- **Simpler Scripts:** Eliminates redundant opcodes like OP_DUP and OP_CHECKSIG.
- **Data Separation:** Moves signatures/public keys to the witness field, reducing ScriptSig size.

Benefits of SegWit Transactions

- **Lower Fees:** Smaller size reduces transaction costs.
- **Scalability:** More transactions per block increase network efficiency.
- **Security:** Fixes transaction malleability by isolating witness data.