Bitcoin Scripting Assignment Report

CS 216: Introduction to Blockchain

Bitcoin Scripting

Team Name: HASHPA

Team Members:

- Lawadya Yashwanth Chowhan (230001046)
- M. Mohan Prudhvi Sai (230001047)
- M. Rishik Preetham (230001048)

Objective

The objective of this assignment is to understand the process of creating and validating Bitcoin transactions using Legacy (P2PKH) and SegWit (P2SH-P2WPKH) address formats. This report details the workflow, transaction details, and comparison of both transaction types using Python and Bitcoin Core RPC.

Part 1: Legacy Address Transactions (P2PKH)

Workflow and Transactions

1. **Setup and Wallet Configuration**
    a. Connected to Bitcoin Core using RPC authentication with:
        i. RPC User: `StandUp`
        ii. RPC Host: `127.0.0.1`
        iii. RPC Port: `18332`
    b. Ensured the wallet "MyNewWallet" exists, loading or creating it if necessary.
    c. Generated 101 blocks using the `generatetoaddress` command to gain initial balance in regtest mode.
2. **Transaction Process**
    a. Generated three legacy addresses:

i. **Address A**: `rpc_client.getnewaddress("", "legacy")`
ii. **Address B**: `rpc_client.getnewaddress("", "legacy")`
iii. **Address C**: `rpc_client.getnewaddress("", "legacy")`

b. Sent 1 BTC from Address A to Address B: `txid1 = rpc_client.sendtoaddress(address_B, 1.0)`

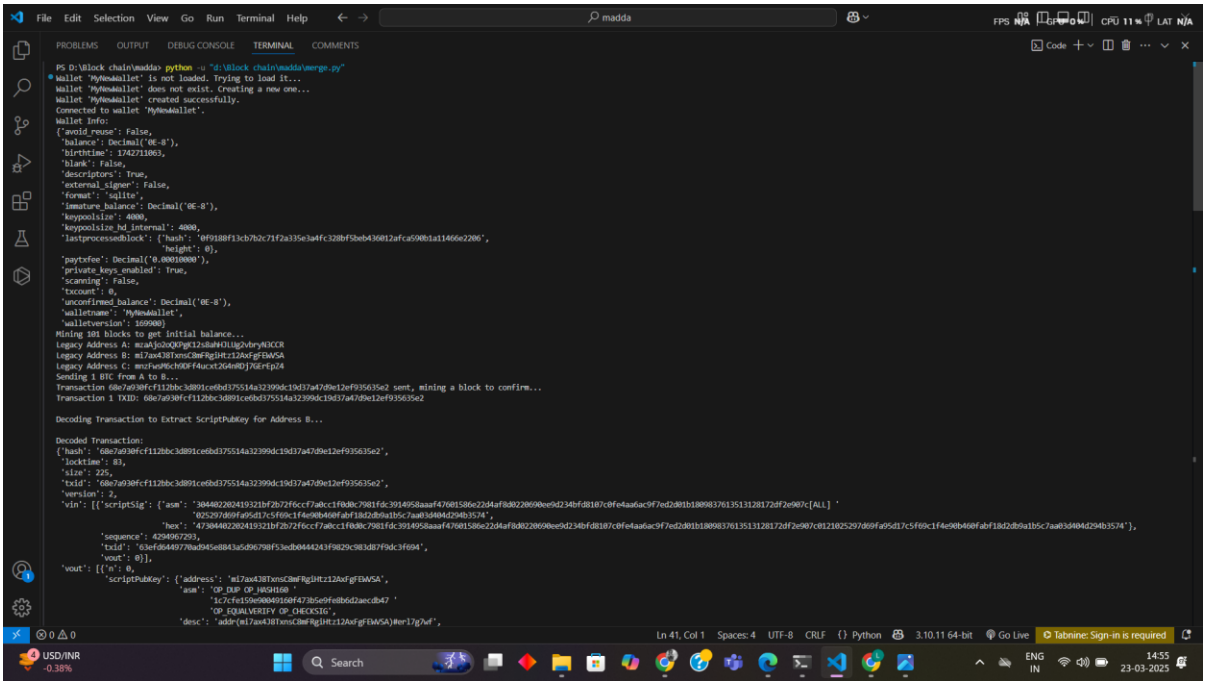
c. Sent 0.5 BTC from Address B to Address C: `txid2 = rpc_client.sendtoaddress(address_C, 0.5)`

d. Decoded the transaction and extracted the locking script (ScriptPubKey) for Address B: `raw_tx = rpc_client.gettransaction(txid1, True)`
`decoded_tx = rpc_client.decoderawtransaction(raw_tx['hex'])`

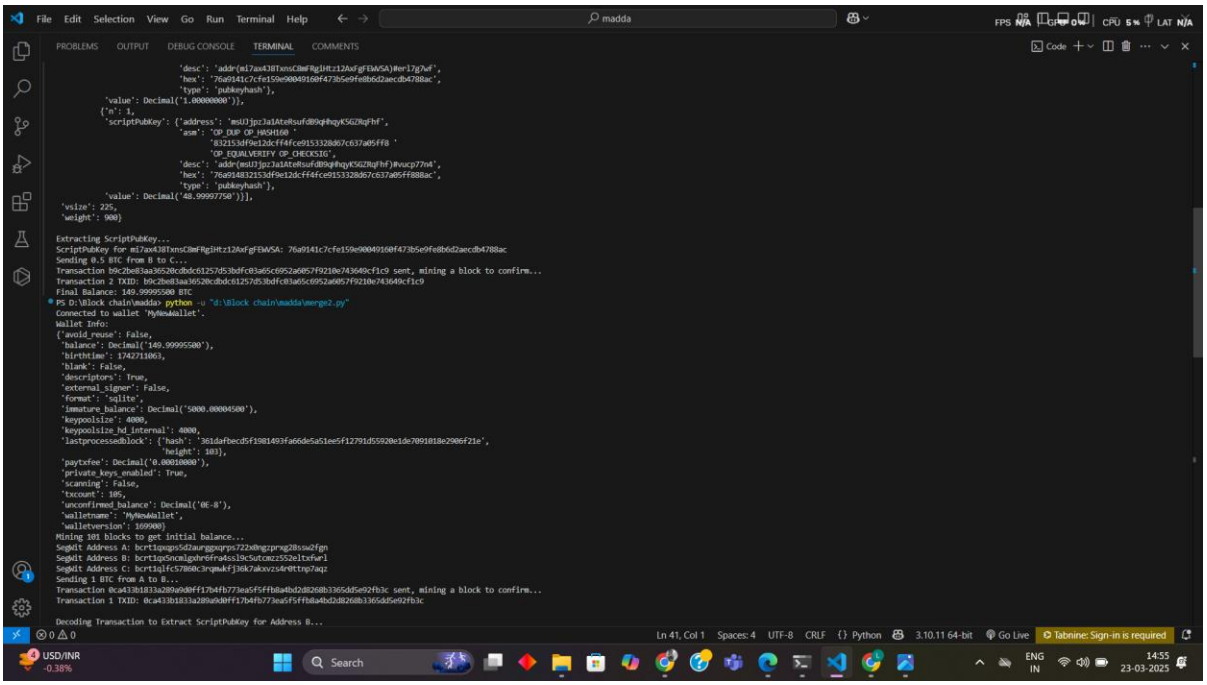
e. **ScriptPubKey for Address B**: mi7ax98TXnsC8mFRgiHTtz12AxFgFEWVSA.

Decoded Scripts

- **Transaction 1 (A → B) Decoded Script**:



- **Transaction 2 (B → C) Decoded Script**:

- 

Challenge and Response Script Analysis

- **Challenge Script (ScriptPubKey)**: This script locks the output using the recipient's public key hash.
- **Response Script (ScriptSig)**: This script contains the unlocking signature and public key, proving ownership.
- **Bitcoin Debugger Validation**: The script execution confirmed that the provided signatures correctly matched the locking conditions.

Part 2: SegWit Address Transactions (P2SH-P2WPKH)

Workflow and Transactions

1. **Wallet Setup and Balance Initialization**
    a. Used the same wallet as in Part 1.
    b. Generated 101 blocks to maintain balance.
2. **Transaction Process**
    a. Generated three SegWit addresses:
        i. **Address A'**: `rpc_client.getnewaddress("", "bech32")`

ii. **Address B'**: `rpc_client.getnewaddress("", "bech32")`

iii. **Address C'**: `rpc_client.getnewaddress("", "bech32")`

b. Sent 1 BTC from Address A' to Address B':

```
txid1 = rpc_client.sendtoaddress(address_B, 1.0)
```

c. Decoded the transaction and extracted the locking script (ScriptPubKey) for Address B'.

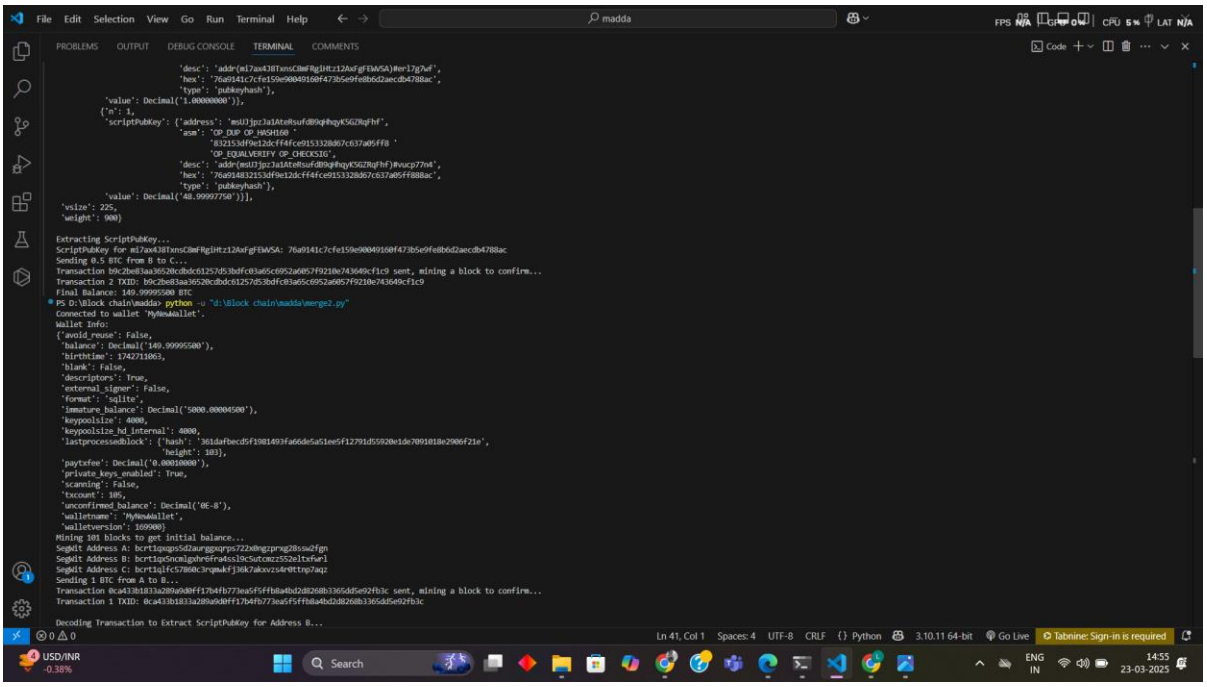
d. Sent 0.5 BTC from Address B' to Address C':

```
txid2 = rpc_client.sendtoaddress(address_C, 0.5)
```

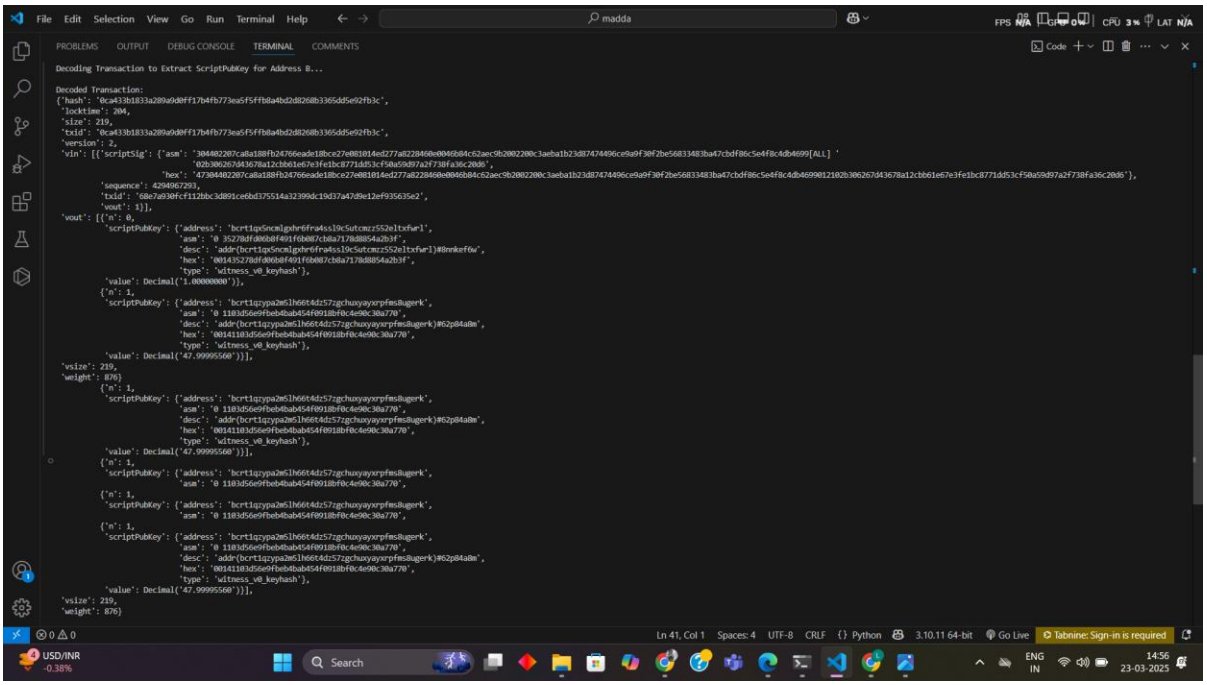
e. Decoded and validated the transaction scripts.

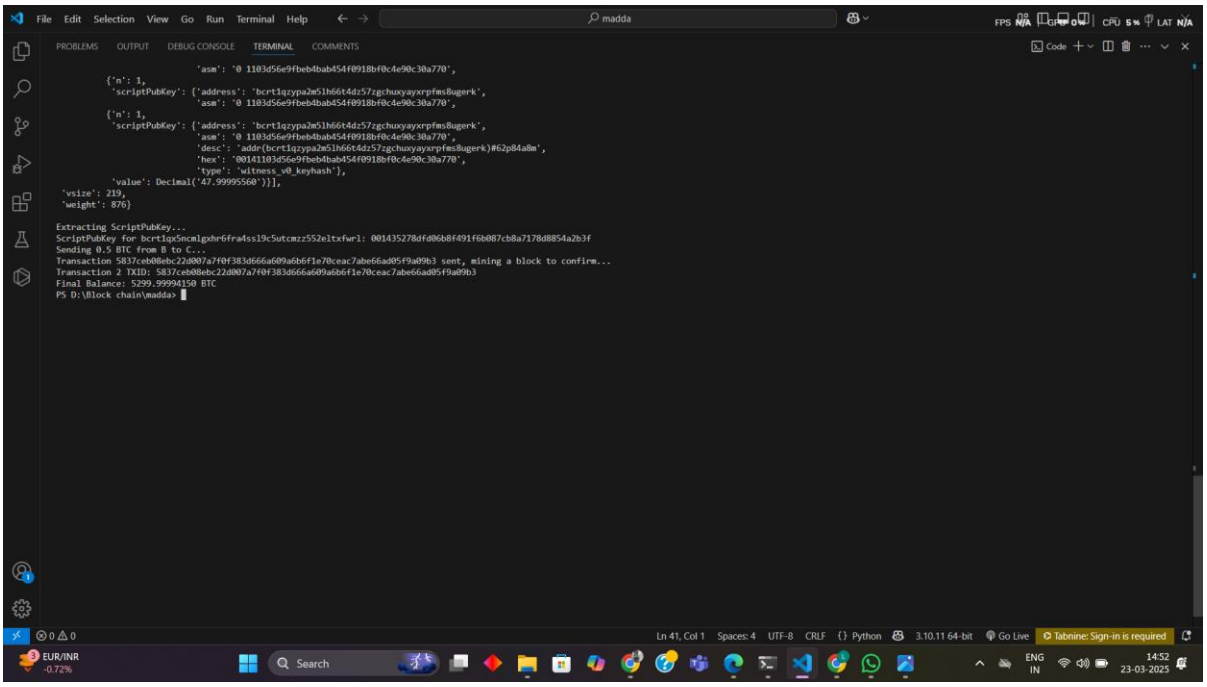
Decoded Scripts

- **Transaction 1 (A' → B') Decoded Script**:



- **Transaction 2 (B' → C') Decoded Script**:

- 

- 

Challenge and Response Script Analysis

- **Challenge Script (Witness Program):** The locking mechanism uses a SegWit script that relies on a separate witness stack.

- **Response Script (Witness Stack)**: The witness data contains the public key and signature for validation.
- **Bitcoin Debugger Validation**: Verified that the witness data correctly satisfies the spending conditions.

Part 3: Analysis and Comparison

Transaction Size Comparison

Transaction Type	Size (vbytes)	Efficiency
P2PKH (Legacy)	Larger (225)	Less efficient
P2SH-SegWit	Smaller (219)	More efficient

Key Differences in Script Structures

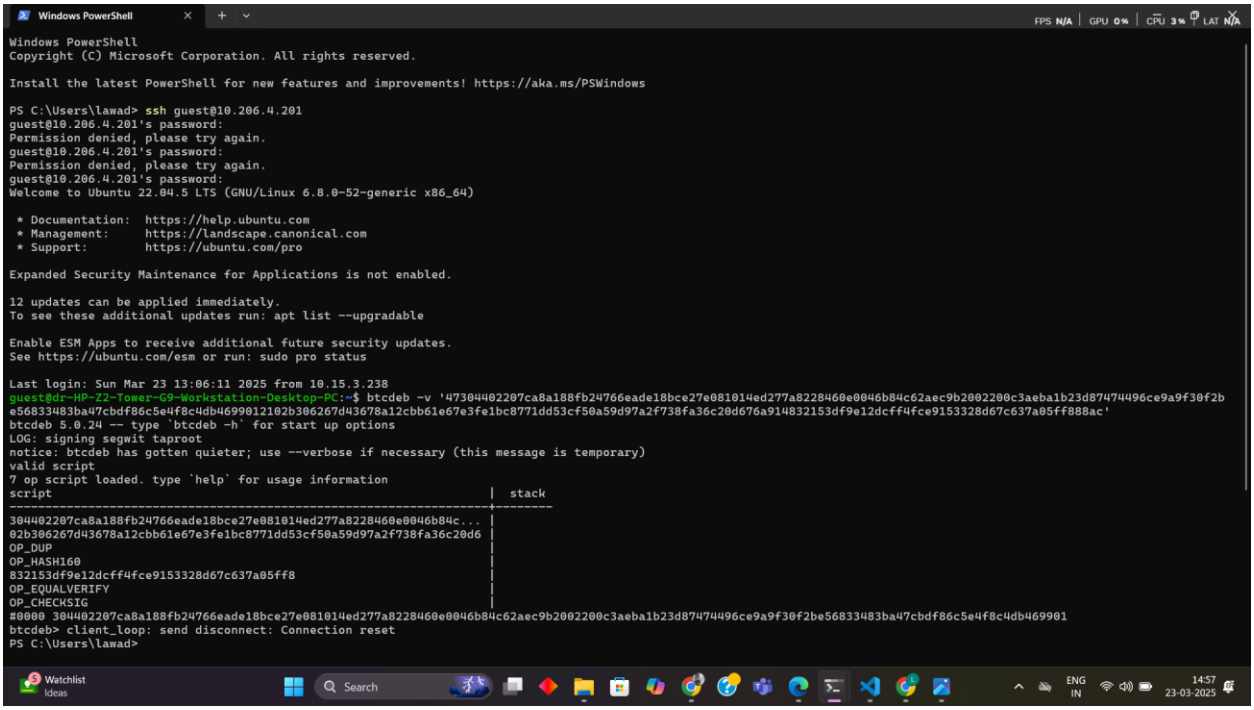
Feature	P2PKH (Legacy)	P2SH-P2WPKH (SegWit)
Challenge Script	ScriptPubKey	Witness Program
Response Script	ScriptSig	Witness Stack
Transaction Size	Larger	Smaller
Fee Efficiency	Higher Fees	Lower Fees

Why SegWit Transactions are Smaller and Their Benefits

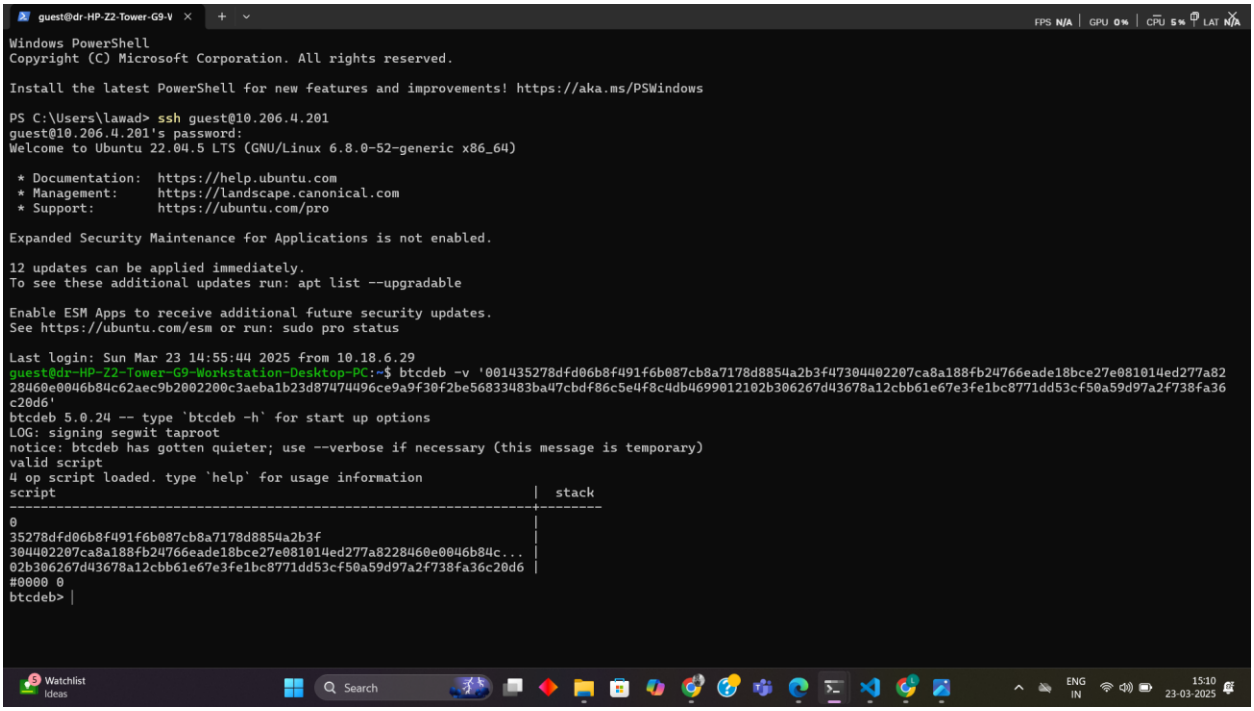
- **Smaller Transaction Size**: SegWit removes the signature data from the main transaction structure, significantly reducing size.
- **Lower Fees**: Due to reduced transaction weight, fees are lower compared to legacy transactions.
- **Fix for Transaction Malleability**: The segregated witness structure prevents TXID modifications, enhancing security and enabling features like the Lightning Network.

Bitcoin Debugging

P2PKH (Legacy)



P2SH-P2WPKH (SegWit)



Conclusion

This assignment successfully demonstrated the creation and analysis of both Legacy (P2PKH) and SegWit (P2SH-P2WPKH) transactions. Through script analysis and transaction comparisons, we observed the efficiency of SegWit transactions over legacy transactions. The use of the Bitcoin Debugger provided deeper insights into the locking and unlocking mechanisms of each transaction type.