Transaction Workflow and Script Analysis Report

Transaction Workflow

Transaction A → B

- The transaction ID is **txid_A_to_B**.
- This transaction is used as input for the next transaction.

Transaction B → C

- The transaction ID is **txid_B_to_C**.
- This transaction uses the **UTXO** (Unspent Transaction Output) from the previous transaction.

Decoded Scripts

Transaction A → B

ScriptPubKey (Locking Script)

php-template:
OP_DUP OP_HASH160 <PubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

- This script locks the transaction output, ensuring only the owner of the corresponding private key can spend it.

Transaction B → C

ScriptSig (Unlocking Script)

- Contains the **signature** and **public key** to unlock the UTXO from **txid_A_to_B**.

Script Analysis

Challenge Script (ScriptPubKey)

- Locks the output requiring a specific public key hash.
- The script ensures that only the owner of the matching private key can unlock and spend the UTXO.

Response Script (ScriptSig)

- Unlocks the output using a **valid signature** and **public key**.

Validation Process

To validate these scripts, use the **Bitcoin Debugger** (available in Bitcoin Core or online Bitcoin script debuggers).

Steps to Run the Code

1. Open a **Bitcoin Debugger** (e.g., Bitcoin Script Debugger).
2. Enter the **ScriptSig** followed by the **ScriptPubKey**.
3. Click **Run** or **Step through** to execute the script.
4. If valid, the script should complete successfully with a `TRUE` output.
5. If invalid, an error will indicate the issue (e.g., incorrect signature or public key).

This process ensures that **txid_B_to_C** correctly spends **txid_A_to_B** following Bitcoin's UTXO model.