from bitcoinlib.wallets import Wallet

from bitcoinlib.transactions import Transaction, Output

from bitcoinlib.services.services import Service

# Step 1: Scan public blockchain for unclaimed, unspent UTXO dust

def scan\_utxo\_dust(service: Service, min\_val=1, max\_val=600):

dust\_utxos = []

utxos = service.utxos()

for u in utxos:

if min\_val <= u['value'] <= max\_val:

dust\_utxos.append(u)

return dust\_utxos

# Step 2: Aggregate multiple dust UTXOs to simulate a "high value" transaction

def build\_phantom\_transaction(dust\_utxos, target\_address):

tx = Transaction(network='bitcoin')

for utxo in dust\_utxos:

tx.add\_input(prev\_txid=utxo['txid'], output\_n=utxo['output\_n'], value=utxo['value'], address=utxo['address'])

tx.add\_output(Output(sum([u['value'] for u in dust\_utxos]), address=target\_address))

tx.fee = 1000 # Minimal fee to look legit

return tx

# Step 3: Sign it with your ghost wallet (pre-prepared key not linked to any identity)

def sign\_and\_simulate\_broadcast(tx, wallet: Wallet):

tx.sign(wallet.keys())

# DO NOT BROADCAST IN REAL WORLD

return tx.as\_dict()