Implementing CoinJoin

Ethan Gordon ‘17 and Patrick Yu ‘16

Basic Idea: What is CoinJoin?

Proposed August 22, 2013 by Gregory Maxwell

http://coinjoin.org

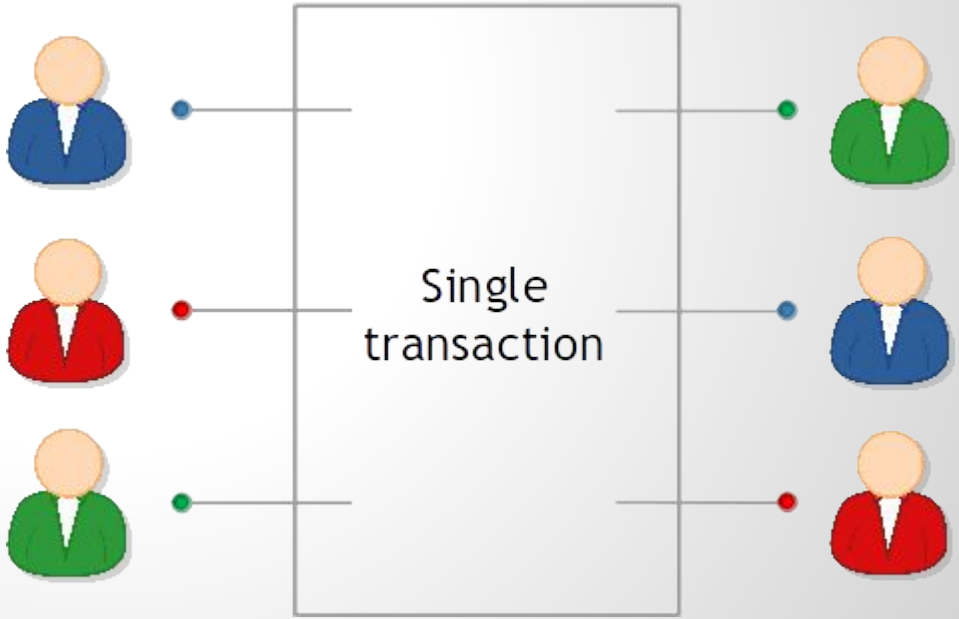


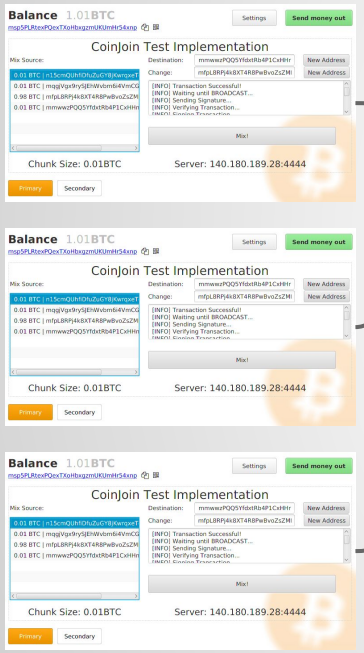
Image from Lecture 6

Current Implementations

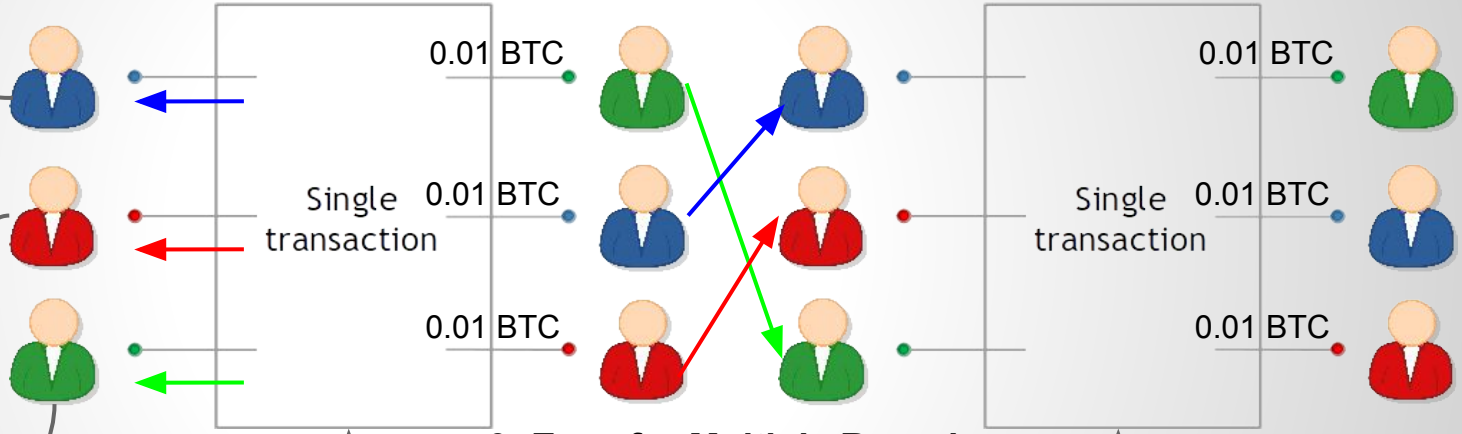
https://github.com/blockchain/Sharedcoin
https://github.com/calafou/coinjoin
https://github.com/maaku/coinjoin

Objectives: 4 Principles of Mixing

1: Automated Client

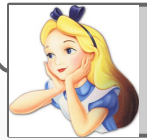


2: Standard Chunk Size



3: Easy for Multiple Rounds

0.001 BTC



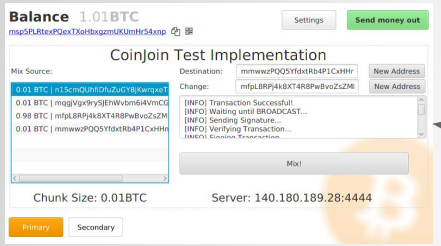
Alice's Altruistic Mixer
Donations Welcome! :)

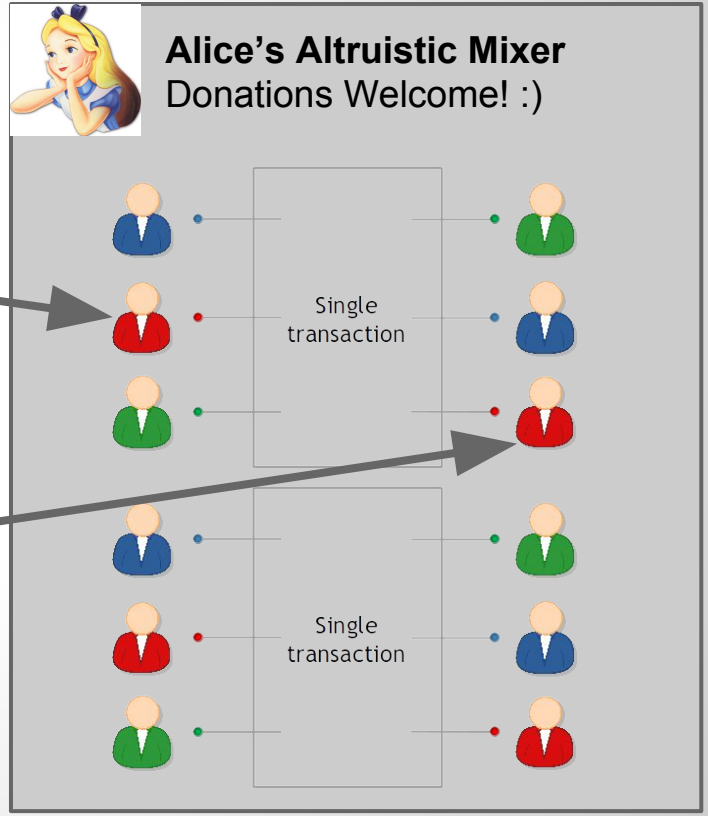
0.001 BTC

4: Fees Are All-or-Nothing (And We Chose Nothing)

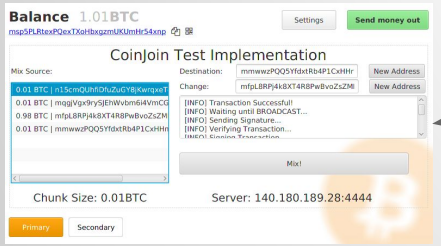
Objectives: Scale and Anonymity

1: Server Handles Multiple Transactions





2: Disconnected Client Interactions



3: Mixer Does Not Know Mapping

Implementation: API Overview

1. getPublicKey()
    a. Parameters: None
    b. Returns: RSA 2048 Public Key
2. registerInput()
    a. Parameters:
        i. Transaction Output with Input Address
        ii. Change Address
        iii. Hash(RSA Public Key)
        iv. Blinded Output Address
    b. Returns: Blinded RSA Signature

Implementation: API Overview

3. registerOutput()
    a. Parameters:
        i. Hash(RSA Public Key)
        ii. Output Address
        iii. RSA Signature
    b. Returns: Full Transaction
4. registerSignature()
    c. Parameters:
        i. Hash(RSA Public Key)
        ii. Input Index
        iii. Input Signature
    d. Returns: TransactionStatus

Implementation: API Overview

5. txidStatus()
    a. Parameters: Hash(RSA Public Key)
    b. Returns: TransactionStatus

Demonstration!

Future Work: Better Fees

1. First Time Fee:
   Require a fee for all non-CoinJoin inputs.
2. Fee Lottery
   a. Commit a value during a transaction.
   b. Have a parallel fee transaction that pays out based on the hash of the committed values.