CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin

Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate

Carolyn Atterbury

CS591 - Bitcoin, Blockchains, and Beyond Spring 2019

Table of contents

Background

Bitcoin

- Public transaction ledger
- Public keys ensure anonymity
- Possible to link PKs to identifying information

Transaction Linking Problem

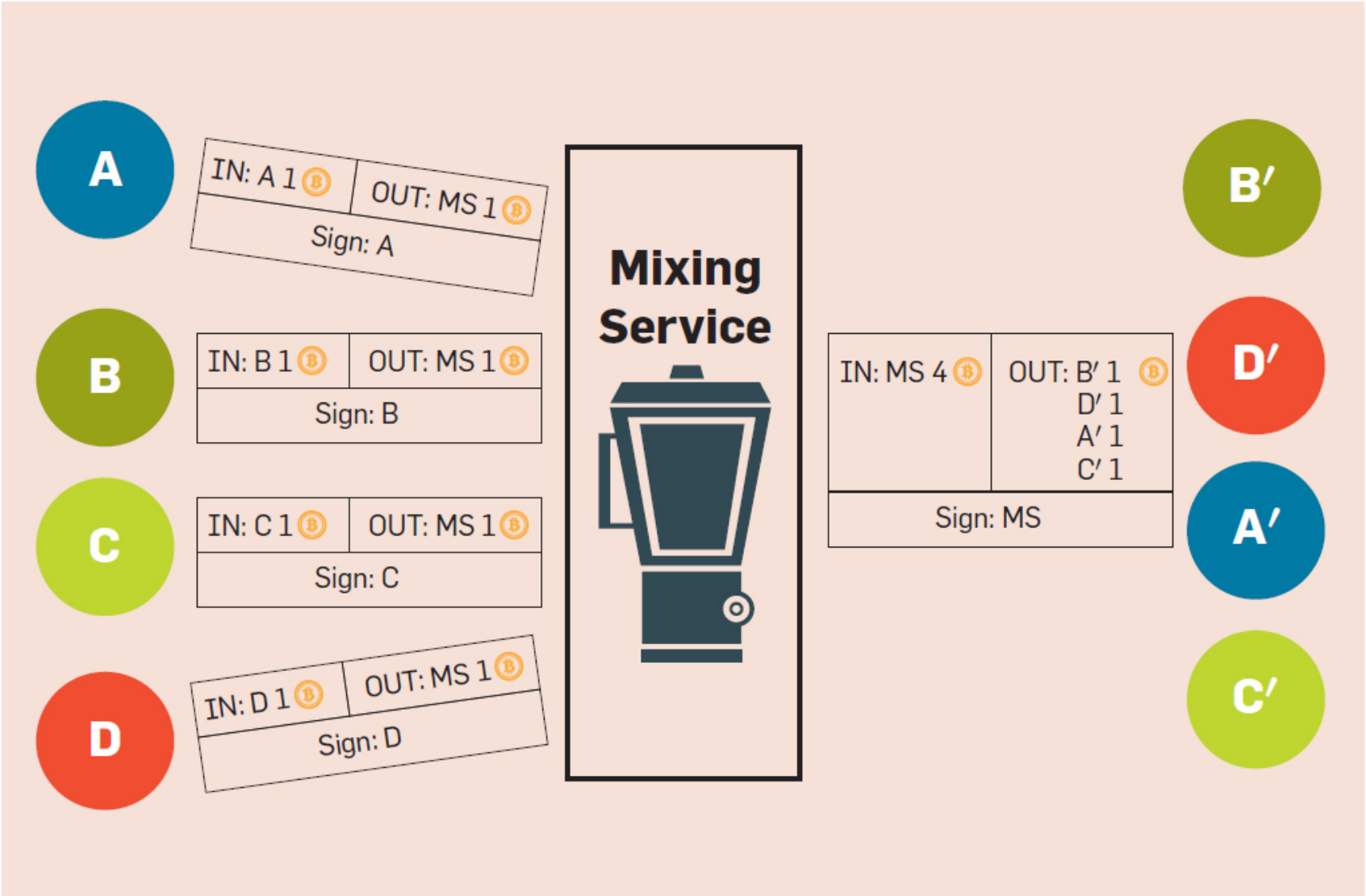
- A person can be linked to their whole transaciton history.

Fungibility Problem

- Not all bitcoins are equal
- Value could change depending on transaction history (e.g. stolen bitcoins, or from the Silk Road)

What is Bitcoin Mixing

- Players hide their transaction in a group of transactions



Mixing Services

MixCoin

- Uses a trusted third party
- Mixing Fee
- Does not preserve anonymity
- User obtains a cryptographic proof of to hold the mix accountable in the case of theft.

ZeroCoin

- Incompatible with Bitcoin.
- Adds overhead and bloat the blockchain.

CoinJoin

- Facilitator shuffles outputs
- Prevents theft, but does not maintain anonymity

Design Goals

Design Goals

- **Unlinkability** - After mixing, input and output address must be unlinkable
- **Verifiability** - An attacker must not be able to steal or destroy a player's coins.
- **Robustness**
- **Compatibility**
- **No Mixing Fee**
- **Efficiency** - Users with limited resources can run the protocol. It should work without waiting for a confirmation from the Bitcoin network.
- **Small Impact on Bitcoin**

CoinShuffle

CoinShuffle

Coinshuffle is completely decentralized, ensures verifiability, anonymity, and robustness against attack.

Users jointly create a single mixing transaction, where:

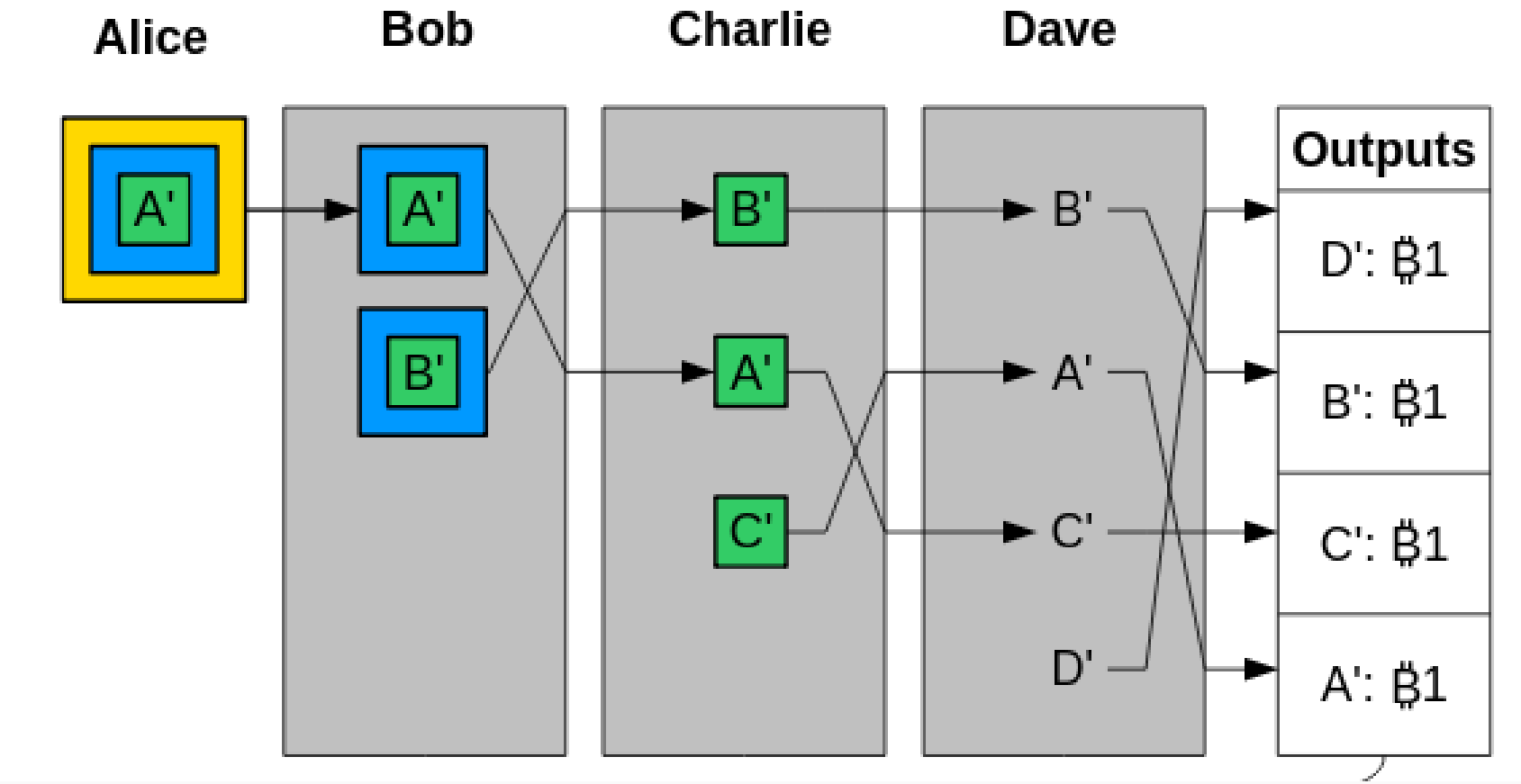
- No double spending
- sum of the input coins = sum of the output coins.
- The transaction must be signed with the private keys corresponding to all the input addresses.

Phase 1 - Announcement

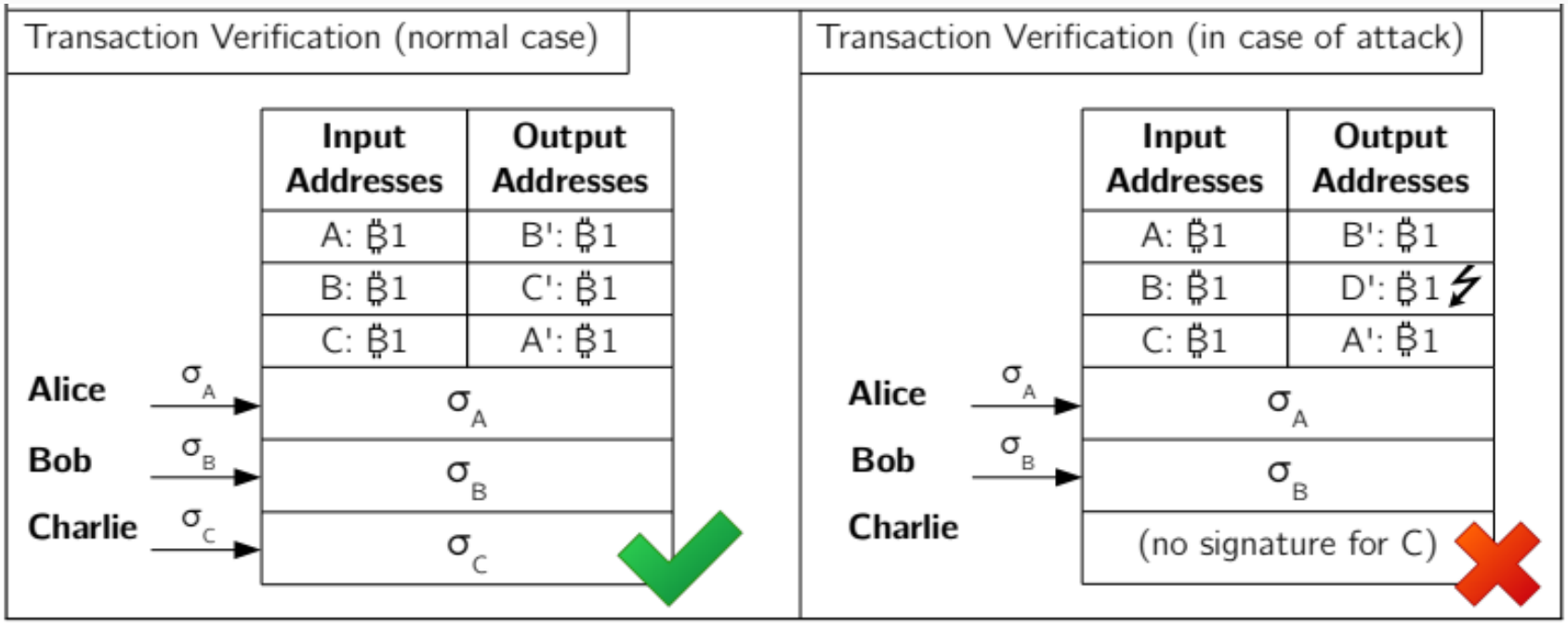
Each player broadcasts:

- Ephemeral public key
- Input address
- Denomination

Phase 2 - Layered Encryption and Shuffling



Phase 3 - Transaction Verification and Blame

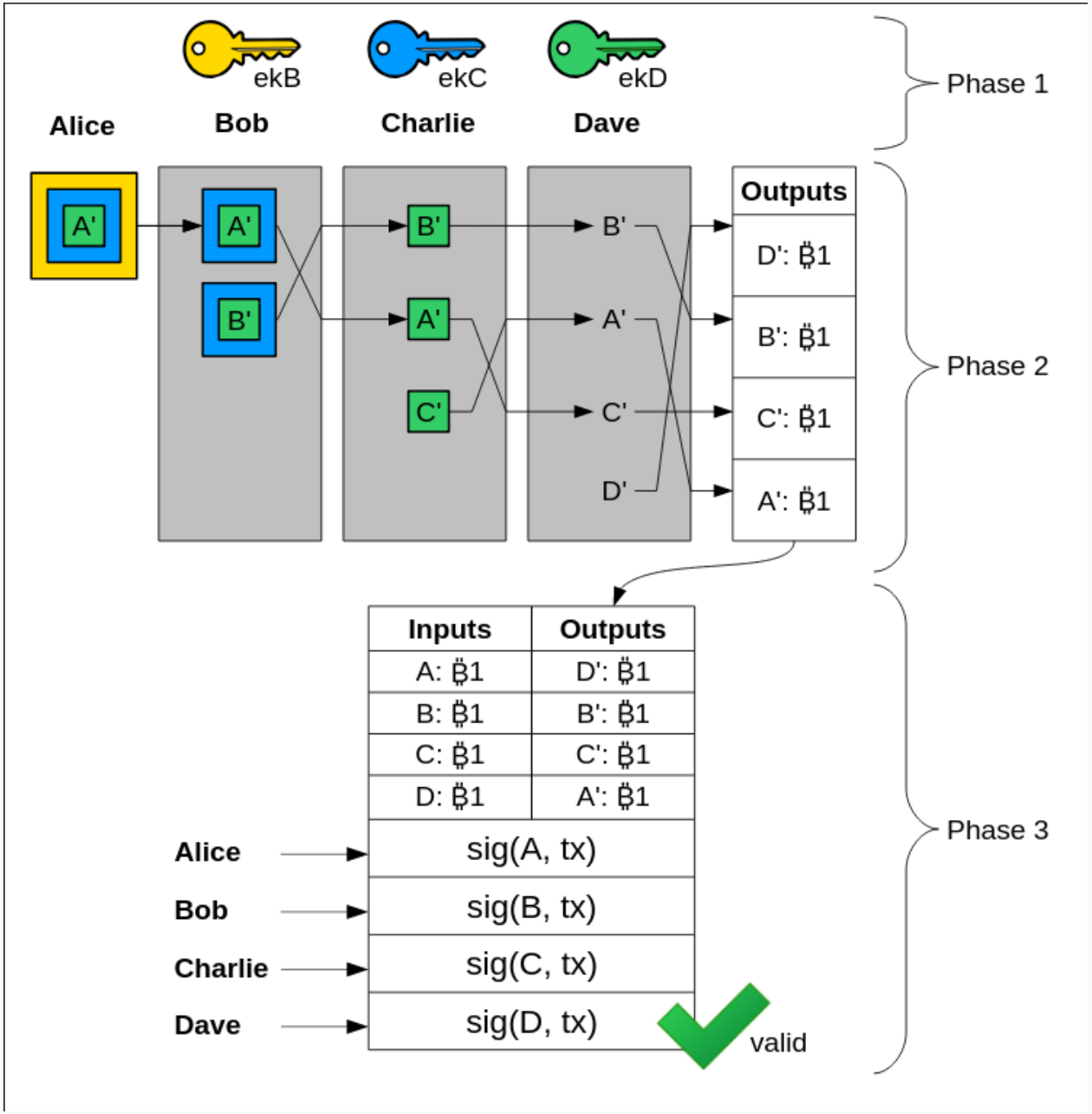


Assigning Blame

Reasons to Blame a user:

- Double spending
- Incorrect layered encryption or shuffling
- A player sends different public keys to different participants

CoinShuffle Protocol



Practical Considerations

Transaction Fees

Transaction Fees can be divided between players

Change Address

Players can have more than one output for the transaction.

Liveness

If someone goes offline, players will wait to hear back from them for a predefined period of time before starting the protocol again without them.

System Discussion and Performance

System Discussion

- **Compatibility** - The protocol not require any change to the Bitcoin system.
- **No Mixing Fee** - There is no additional mixing fee for running the protocol.
- **Efficiency** - The protocol used standard public key encryption, so users with computationally restriced hardware can run the protocol.
- **Small Impact on Bitcoin** - The protocol creates only one mixing transaction.

Performance

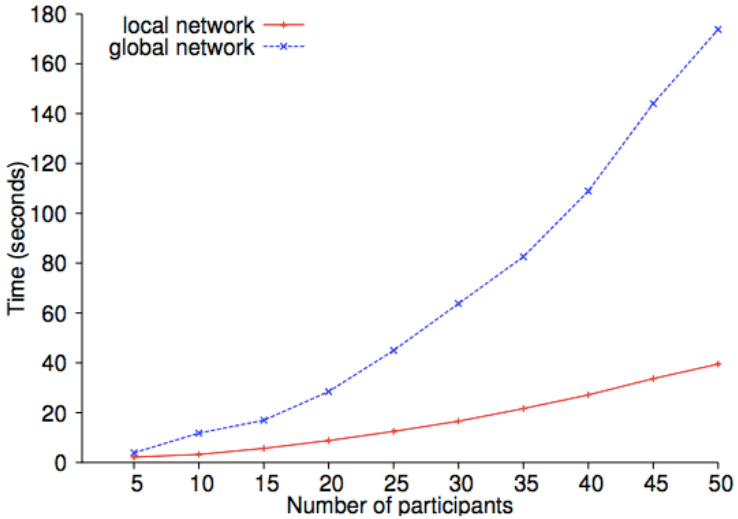


Fig. 3. Overall execution time

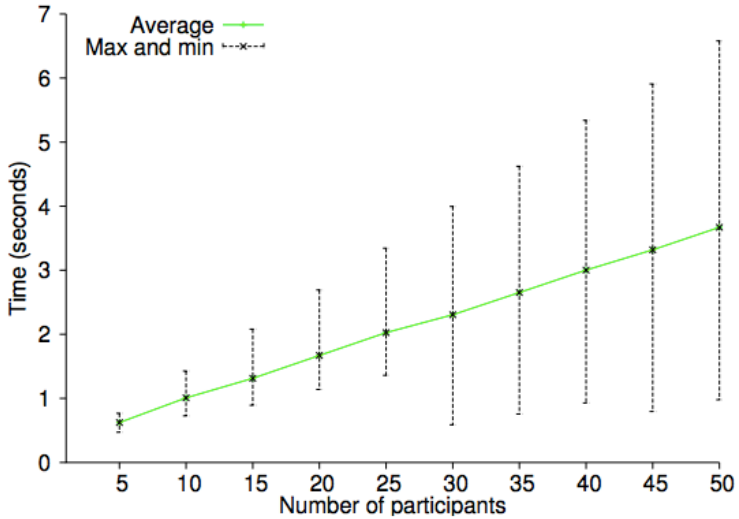


Fig. 4. Processing time per node

Current Implementations

- CashShuffle
- ShufflePuff

Coinshuffle ++ is the successor of CoinShuffle. (much faster)

Questions?