# CoinShuffle

PRACTICAL DECENTRALIZED COIN MIXING FOR BITCOIN

# CoinShuffle (2014)
# Ruffing, Moreno-Sanchez and Kate

**CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin***

Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate

MMCI, Saarland University
{tim.ruffing,pedro,aniket}@mmci.uni-saarland.de

https://petsymposium.org/2014/papers/Ruffing.pdf

# Last week – SNICKER Coinjoin

Summary of SNICKER CoinJoin idea:

**Non-interactive CoinJoins** are possible between two participants if the **proposer assumes likely UTXOs** and tweaks a revealed public key with a Diffie-Hellman shared secret. PSBT of this form can be broadcast to a public forum and then signed when possible by the other party.

# Wasabi Research Club

- January 6th, 2020 – Knapsack CoinJoin
- January 13th, 2020 – SNICKER
- January 20th, 2020 – CoinShuffle
- January 27th, 2020 – TBD
  https://github.com/zkSNACKs/WasabiResearchClub

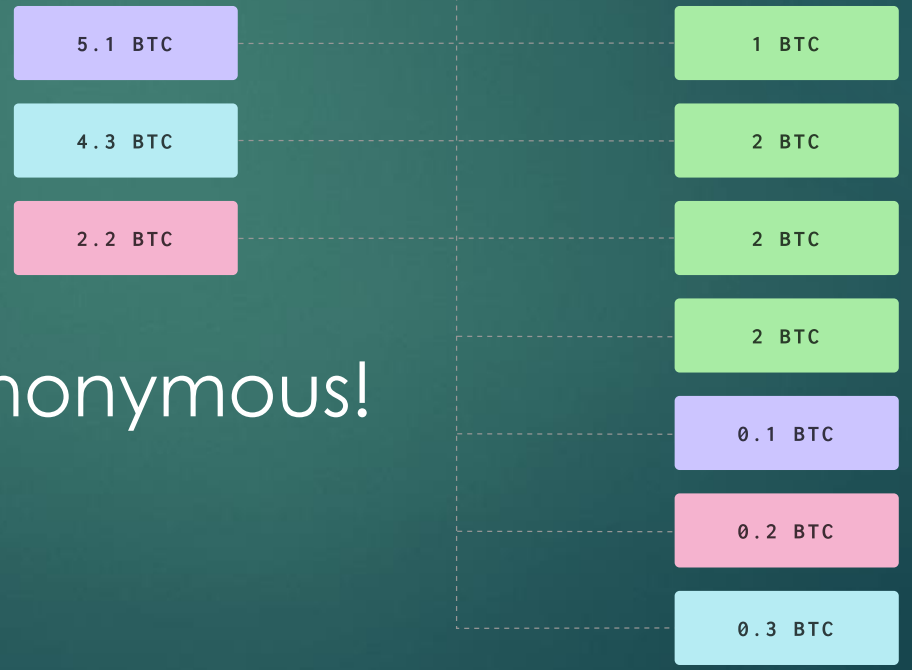
# Problem with current CoinJoins

- Interactive (requires a server)
  - Could reduce the privacy of users
  - Difficult to coordinate many participants
  - Fragile to attack (central point of failure)

# Could a CoinJoin be done *without* a central Coordinator?

What requires coordination?

- Inputs
- Outputs*
- Signatures




- *Output must be anonymous!

# Review of Wasabi Coordinator

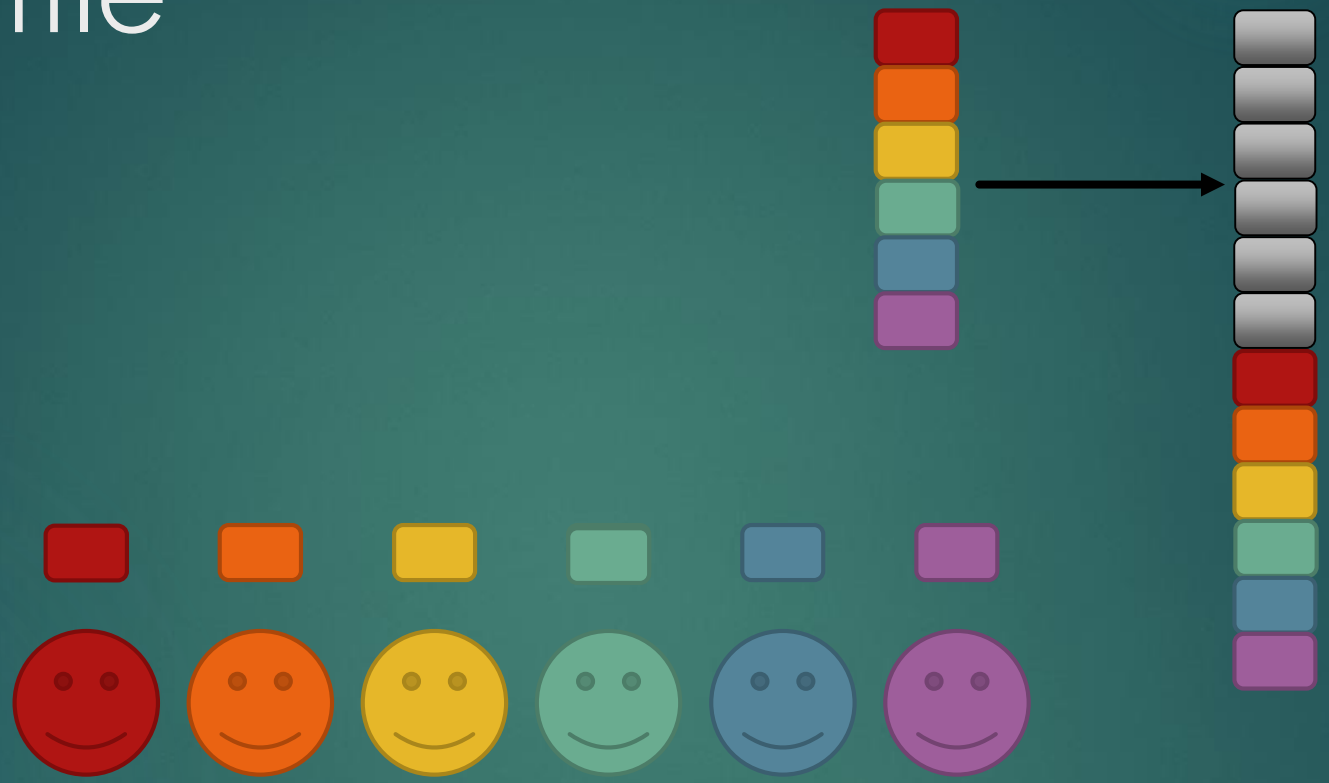
- Step 1 – Register Inputs, change outputs and blinded output
- Step 2 – Server accepts and returns blind signed output
- Step 3 – Connection confirmed at time of CoinJoin
- Step 4 – Outputs are unblinded and posted to the Server
- Step 5 – CoinJoin Transaction is constructed and given to participants
- Step 6 – Signed by all participants and returned to server
- Step 7 – Signatures are collected and a signed CJ transaction is broadcast

# CoinShuffle – Wasabi without the coordinator

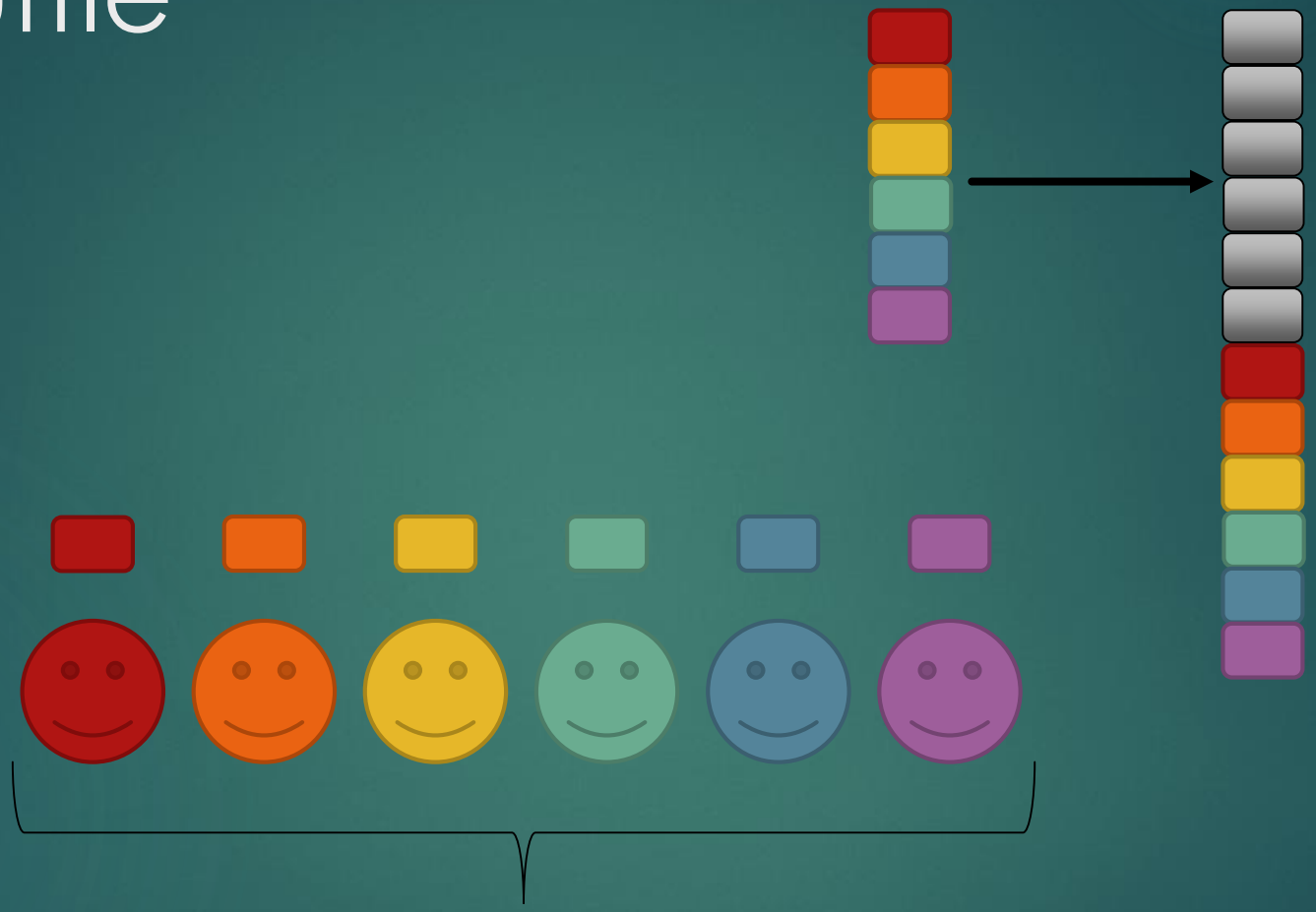
- Using DISSENT protocol for communicating anonymous outputs by participants.

# CoinShuffle

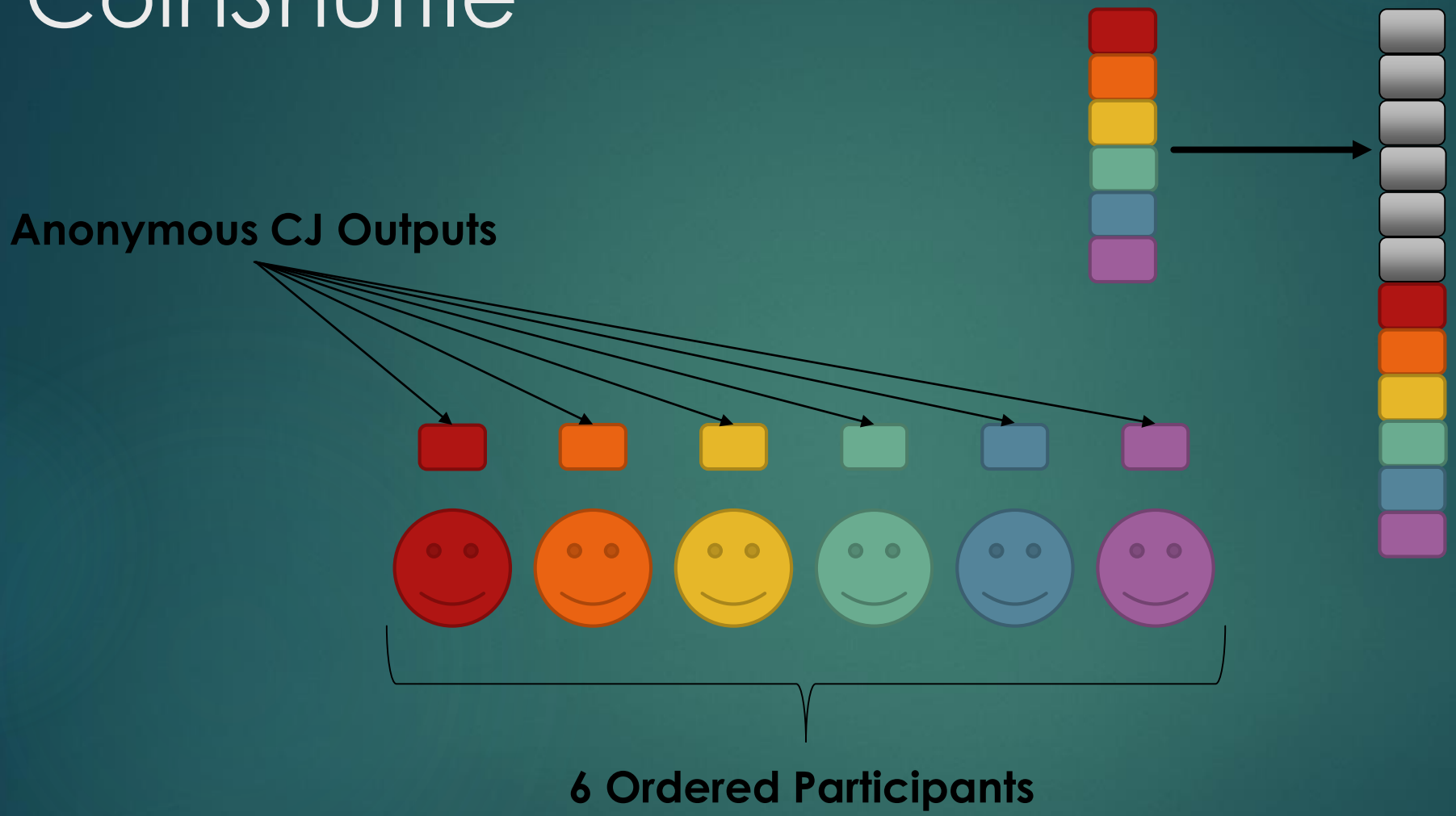
# CoinShuffle



# CoinShuffle

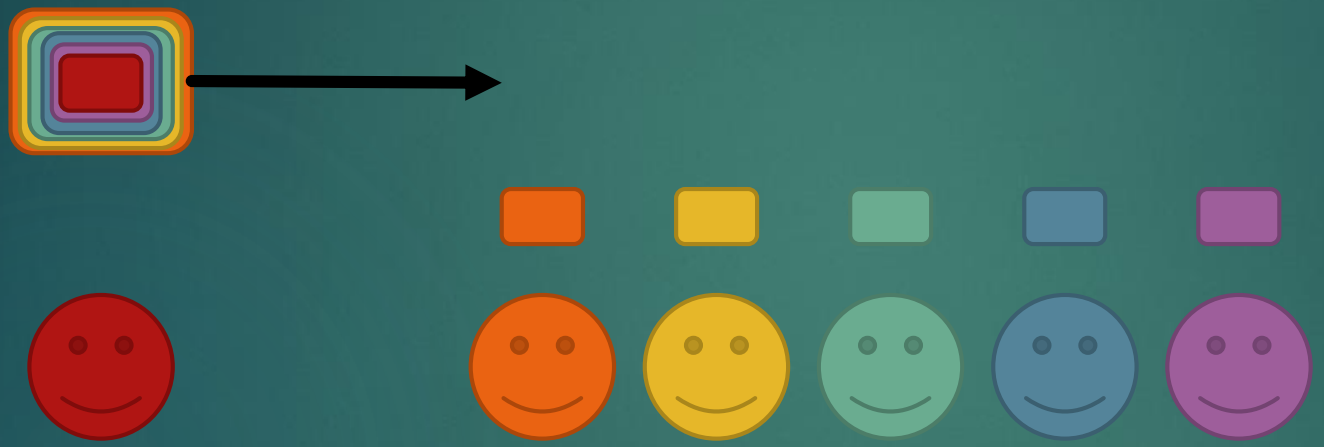


# CoinShuffle

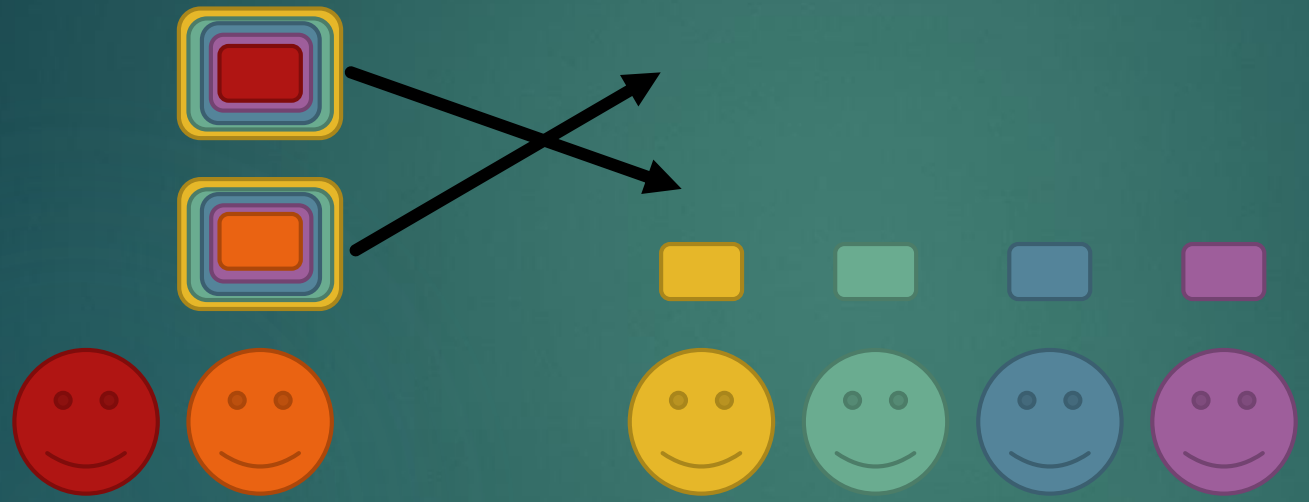


# CoinShuffle

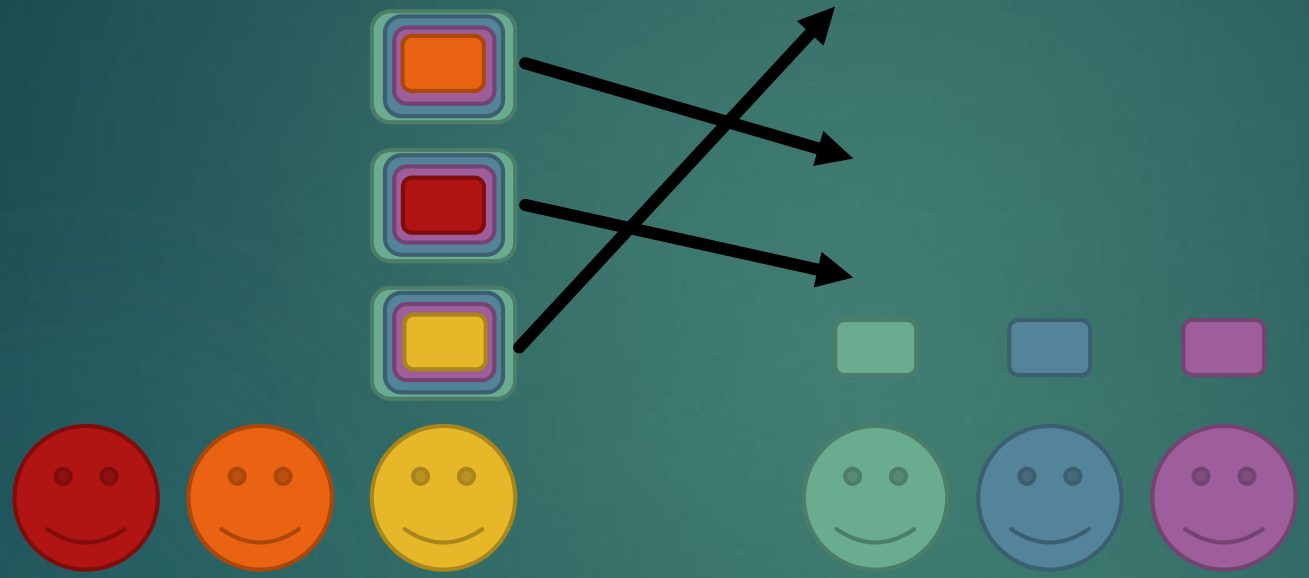
# CoinShuffle



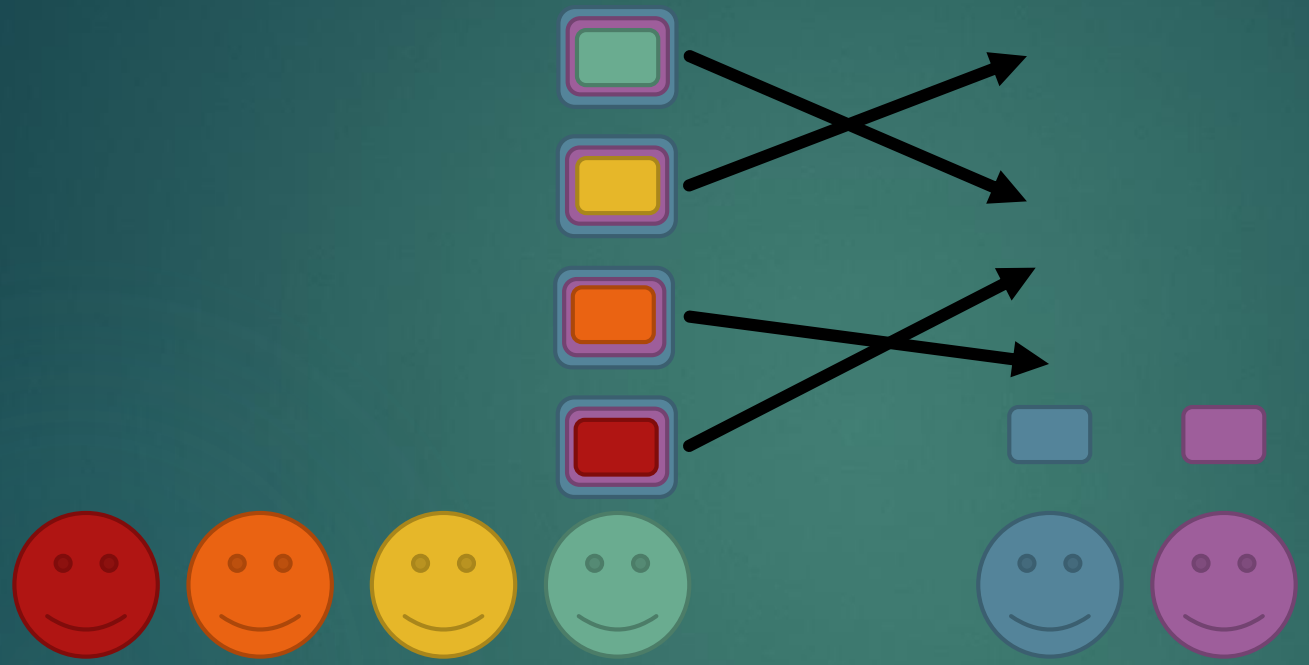
# CoinShuffle



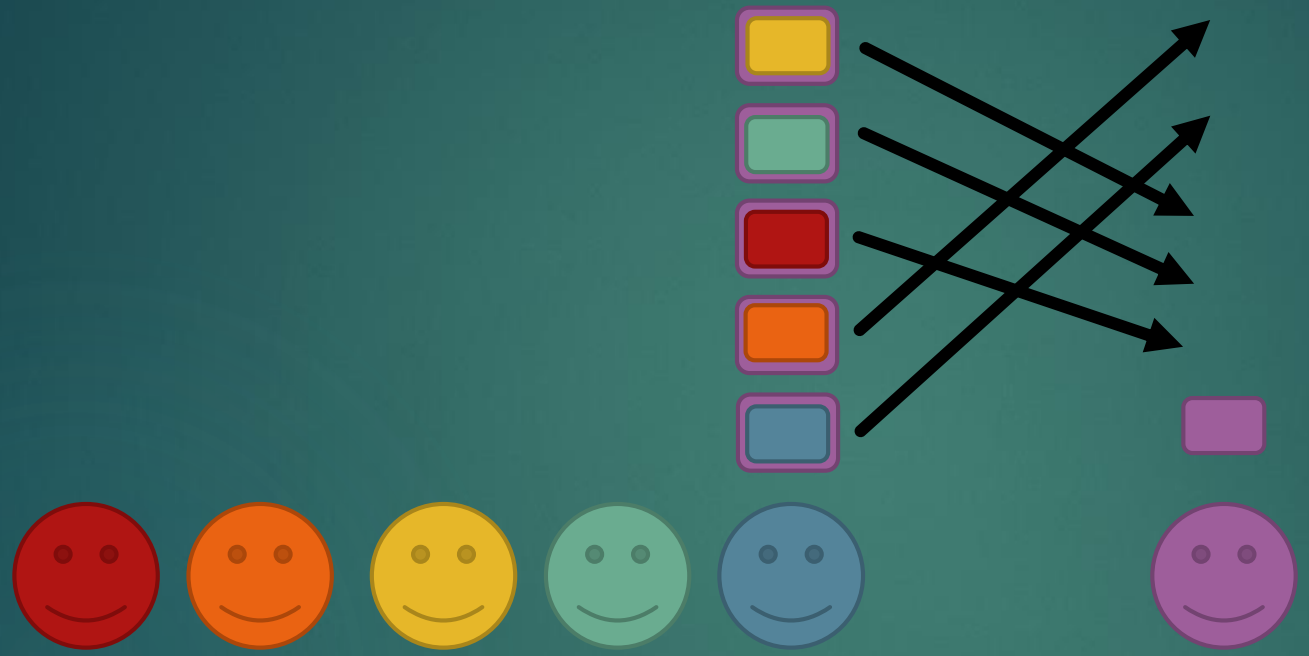
# CoinShuffle



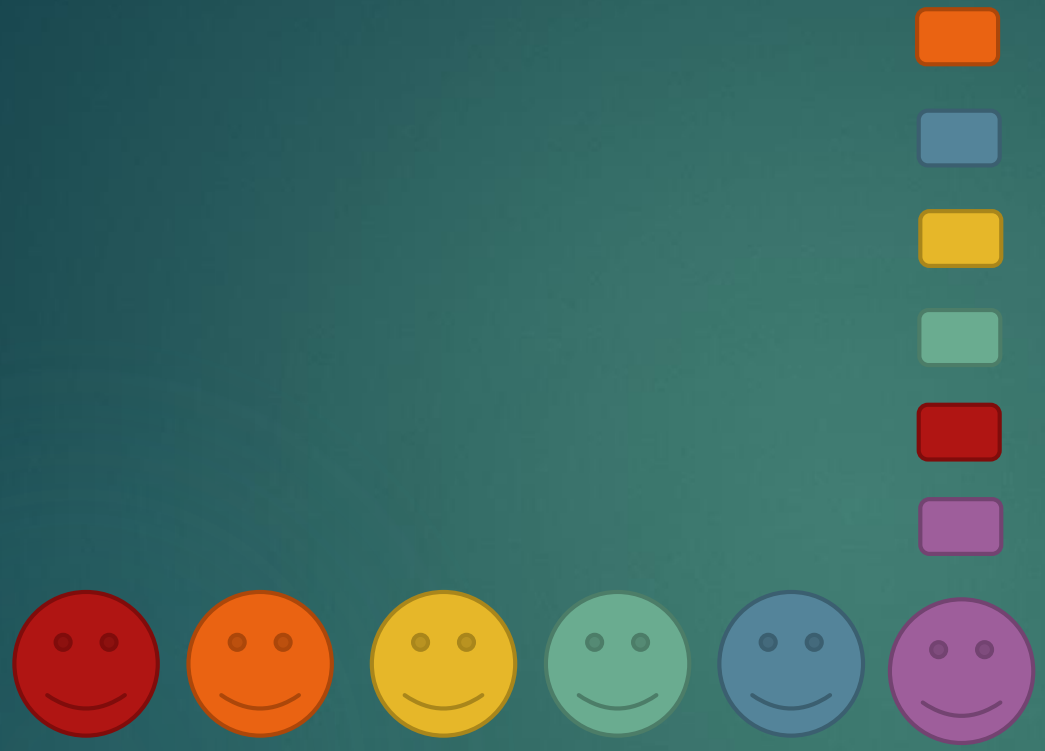
# CoinShuffle



# CoinShuffle

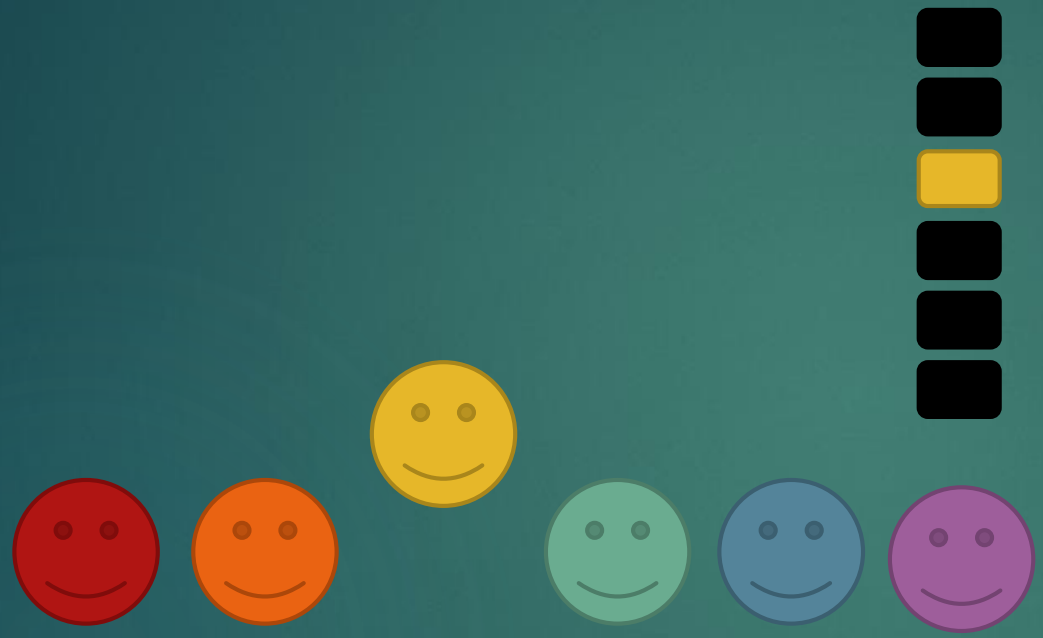


# CoinShuffle



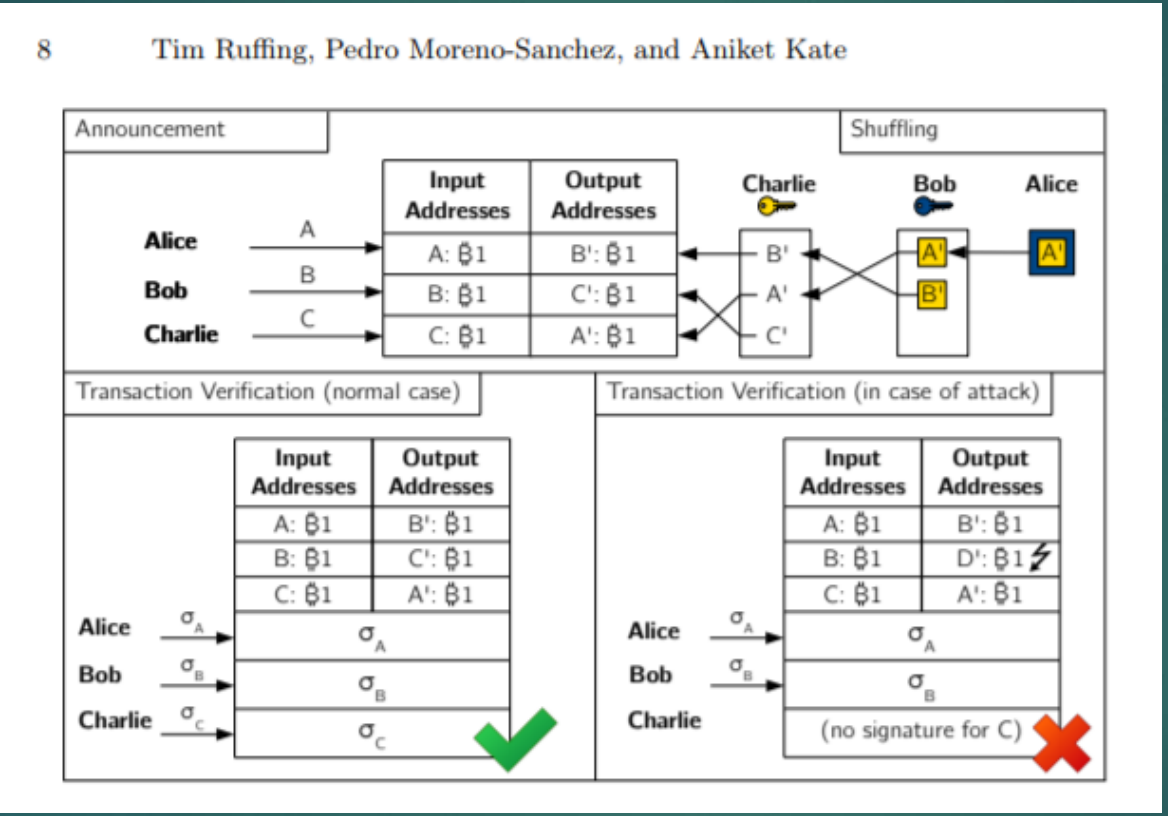
# CoinShuffle

# CoinShuffle



# CoinShuffle

# CoinShuffle



# CoinShuffle - Performance

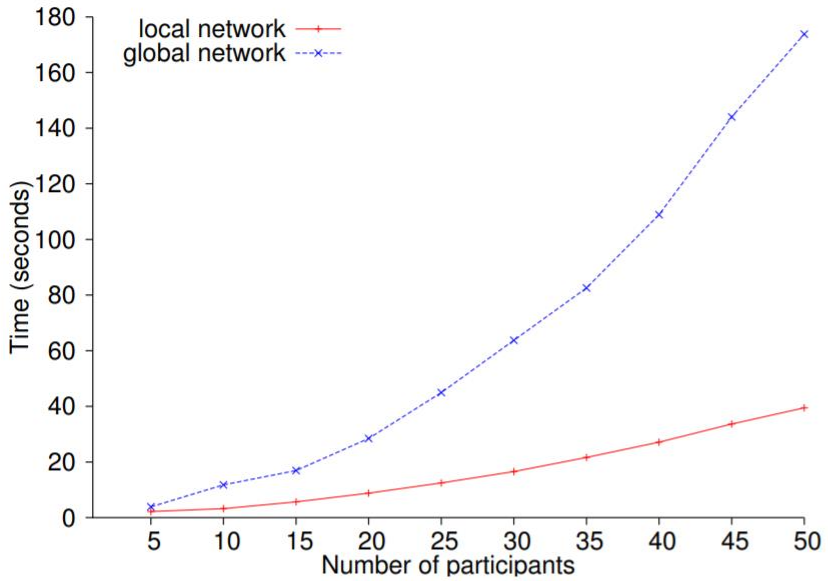



**Fig. 3.** Overall execution time

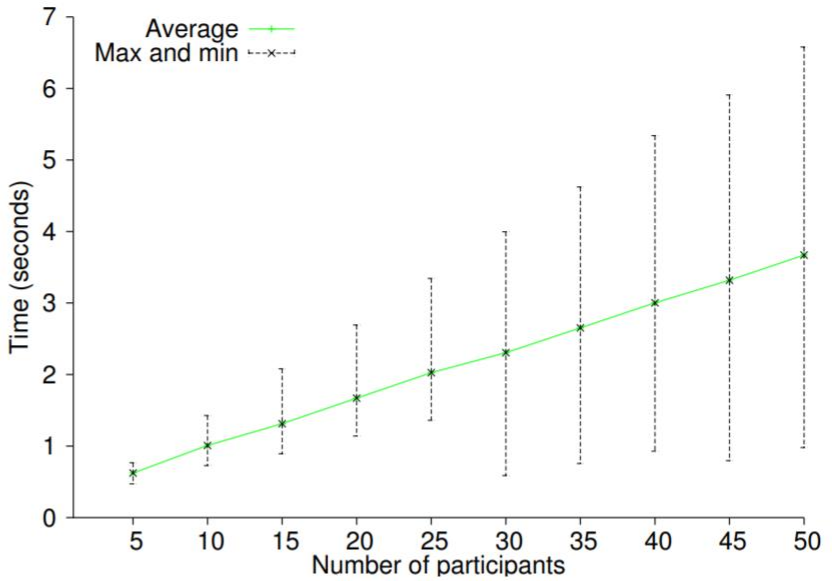
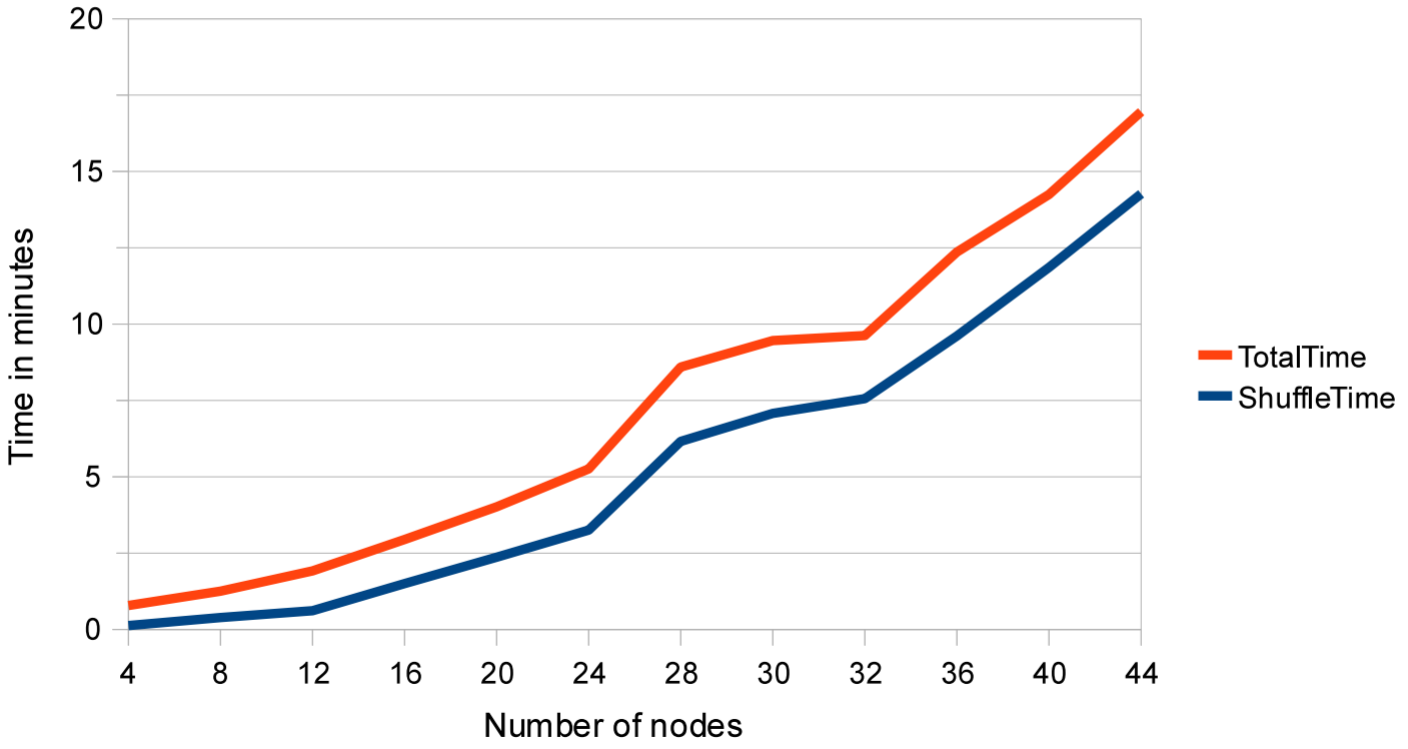



**Fig. 4.** Average processing time per node

# Dissent - Performance

**Figure 4: Time required to send varying message sizes, broken into shuffle and bulk transfer protocol portions.**



# CoinShuffle - Summary

- Rather than have a Secure Multi-Party Computation with a coordinator, CoinShuffle aims to solve the problem of constructing a CoinJoin with just the participants themselves.
- Using the Dissent messaging protocol, CoinShuffle participants shuffle their anonymous outputs until all outputs are made available to all participants, without a link from any participant to an output.
- Biggest drawback is the time cost as number of participants grow
- Biggest advantage is that there is no coordinator to DOS

# CoinShuffle - Discussion