CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin

[project page]
[paper]

Tim Ruffing
@real_or_random

Pedro Moreno-Sanchez
@pedrorechez

Aniket Kate
@aniketpkate

MMCI, Saarland University

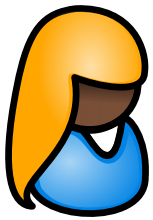
CLUSTER OF EXCELLENCE

ESORICS 2014

UNIVERSITÄT DES SAARLANDES

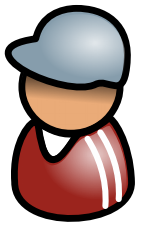
Introduction

Alice



Money

Dealer



Hash(vk)
= Bitcoin address

`3J98t1WpEZ73CNmQv`
`iecrnyiWrnqRhWNLy`

Bitcoin Transaction

Bitcoin 101

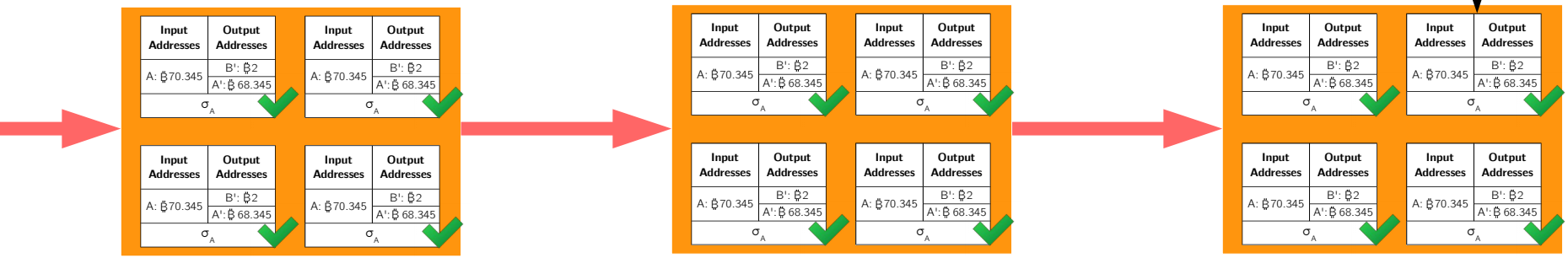
A: Bitcoin Address
A = Hash(vk)

sk: signing key
vk: verification key

Bitcoin Transaction

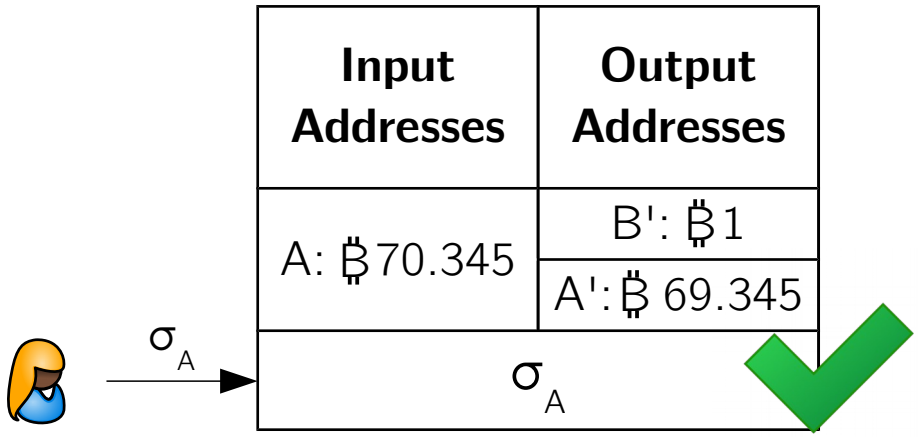
σ_A

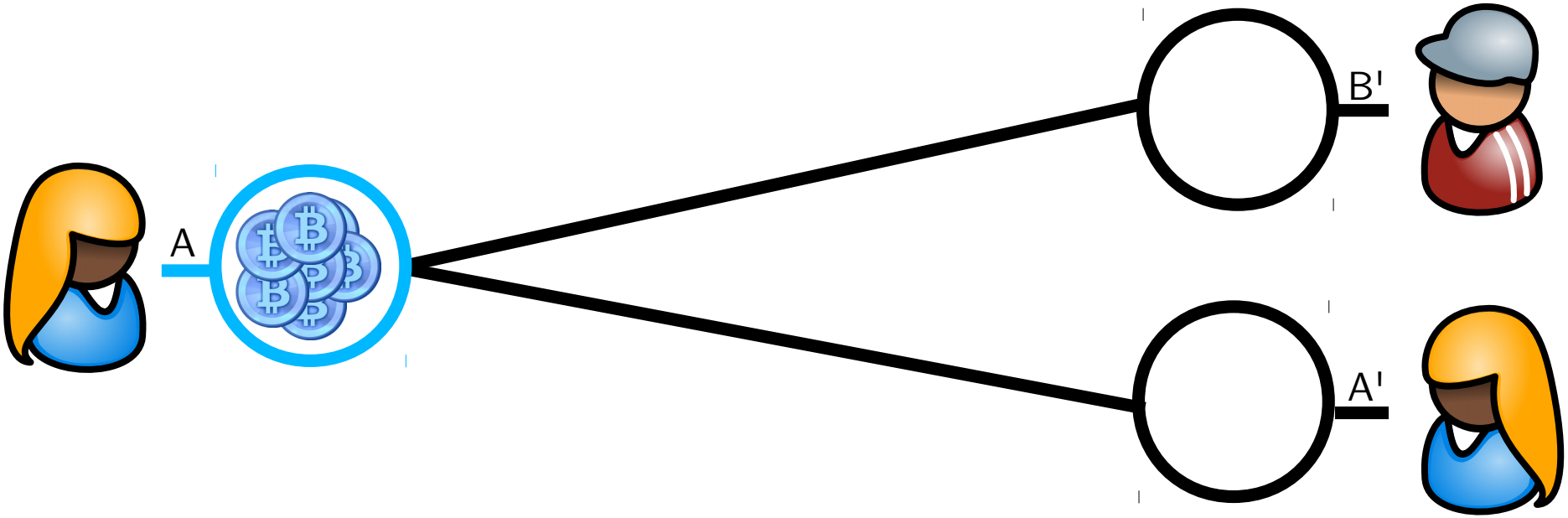
Input Addresses	Output Addresses
A: ฿70.345	B': ฿1
	A': ฿ 69.345
σ_A	



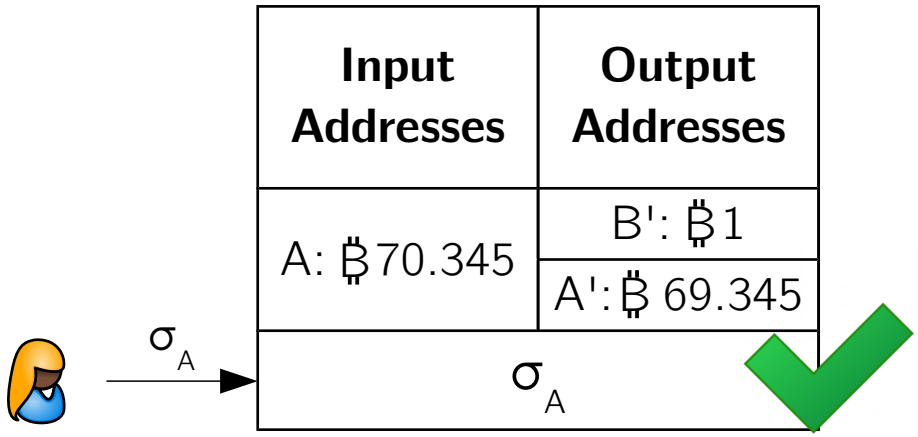
public list of transactions

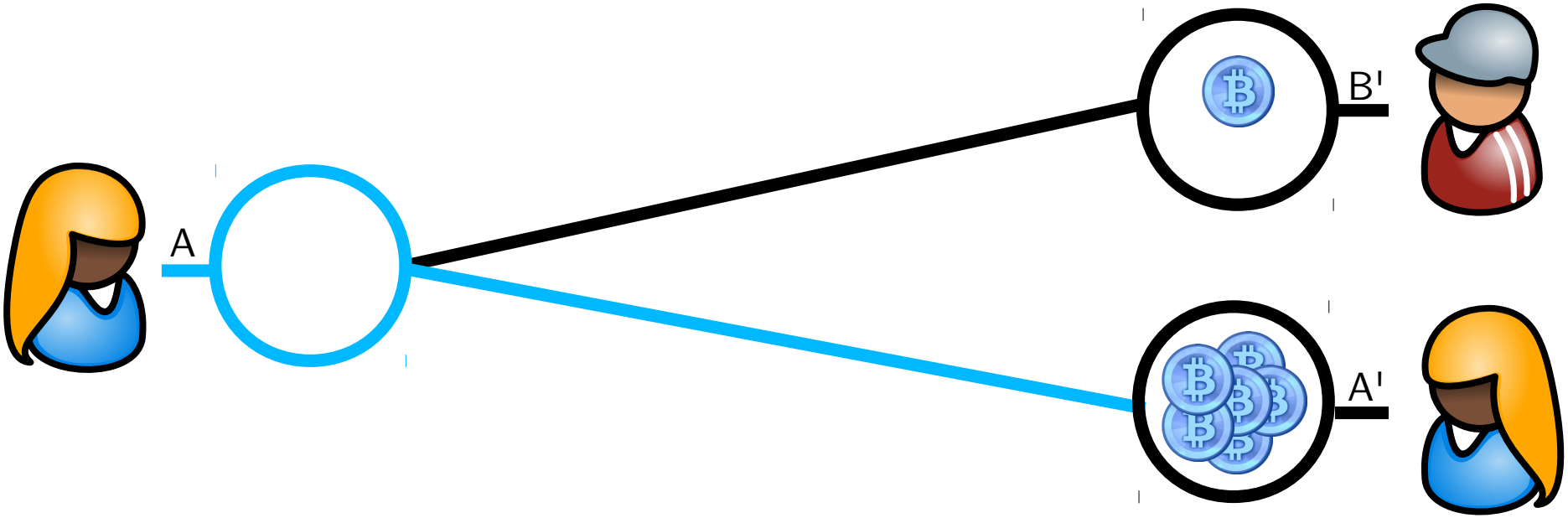
Linkability of Pseudonyms



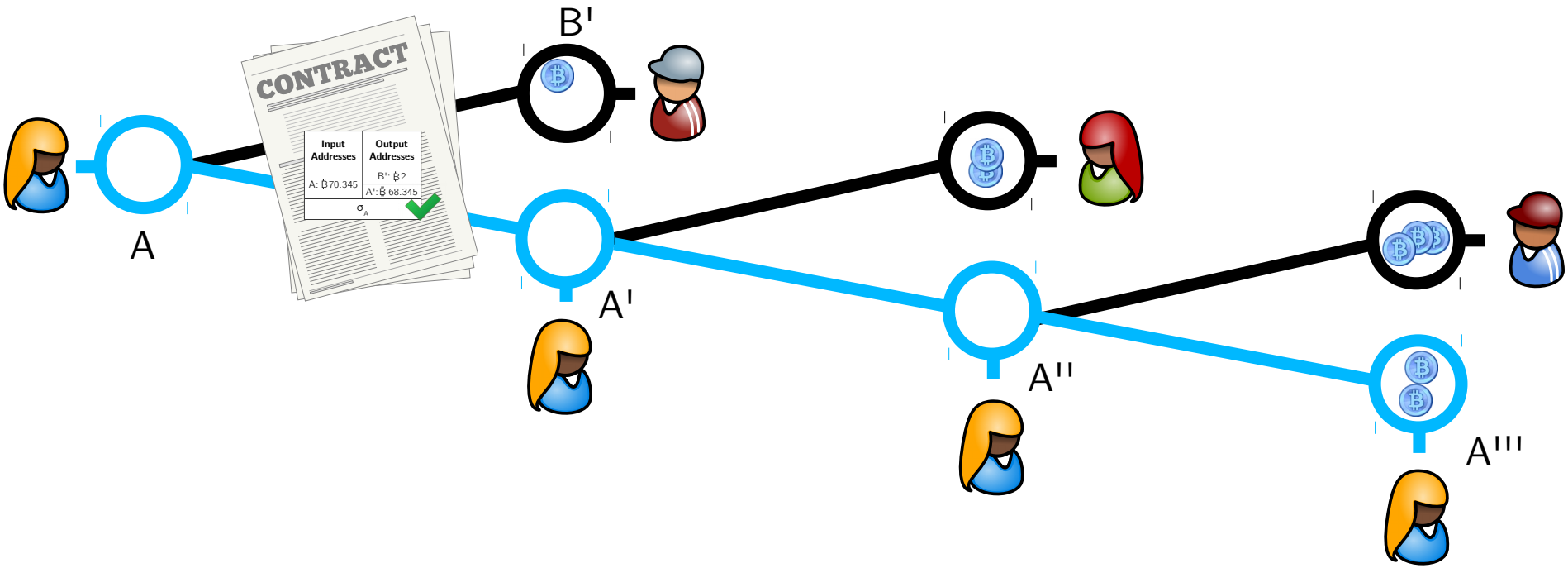


Linkability of Pseudonyms

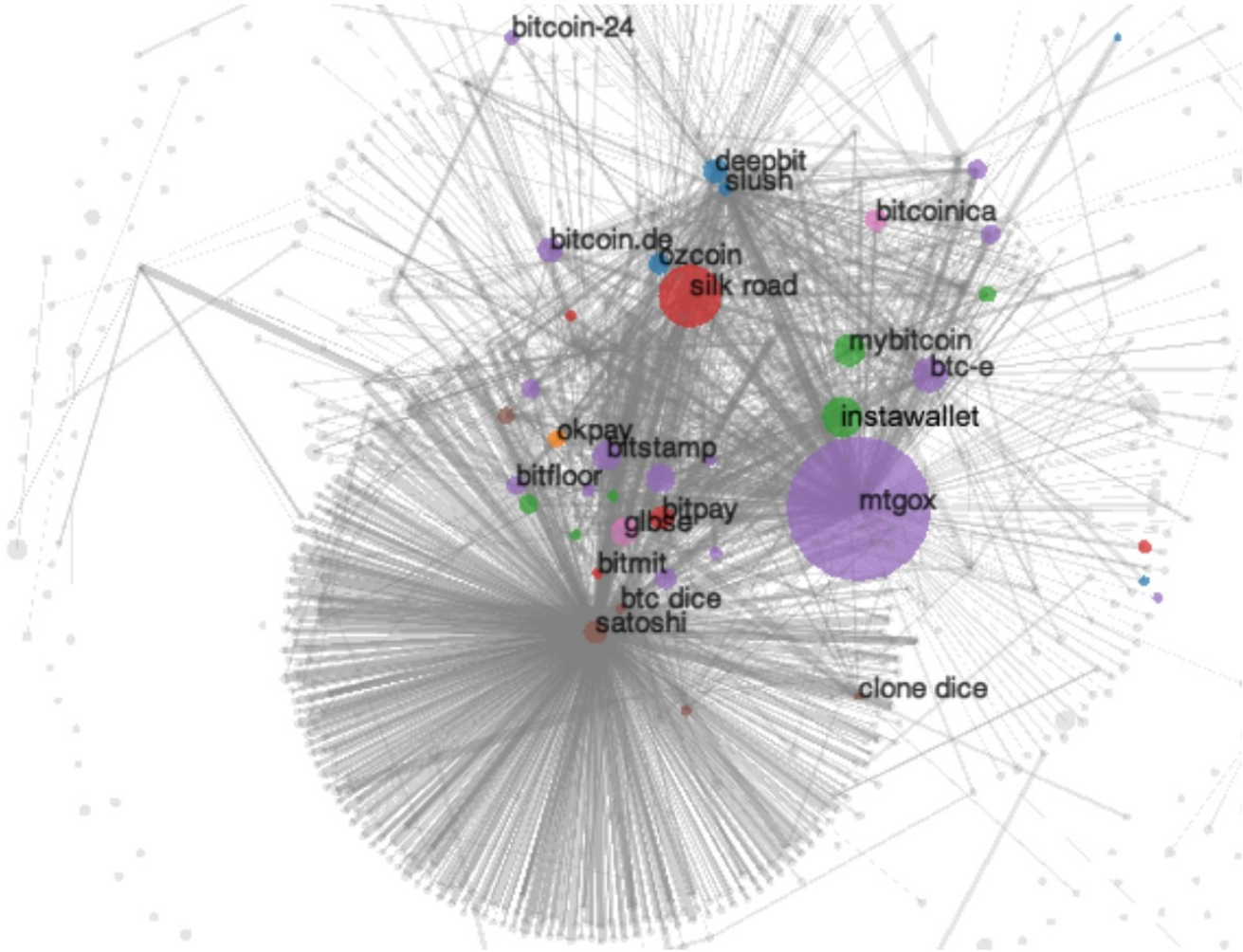




Linkability of Pseudonyms



Deanonymization Attacks in Practice



[Meiklejohn et al., IMC'13]

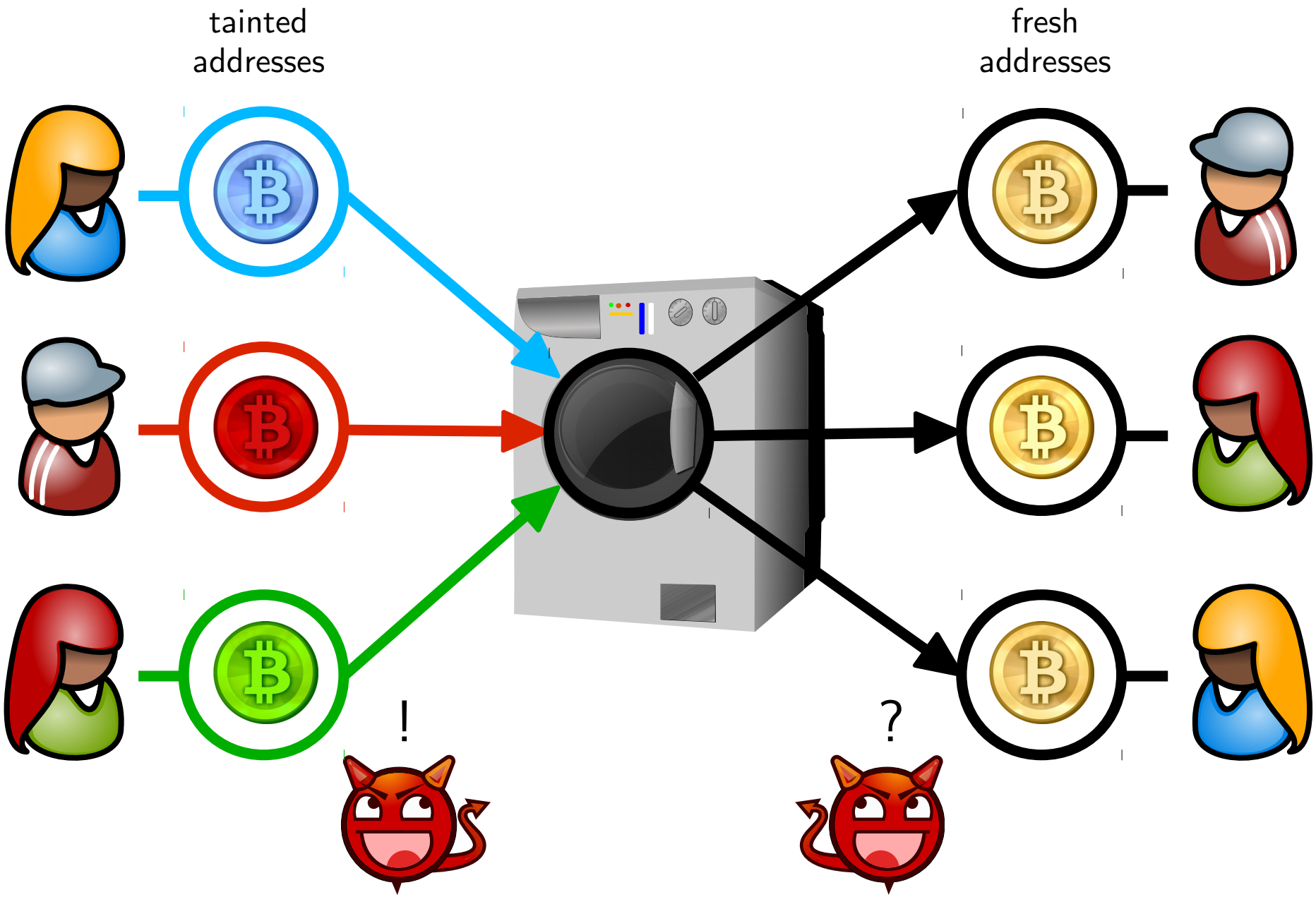
Push-the-button tool for clustering:
Check out bitiodine.net [Spagnuolo et. al., FC'14]

Avoiding the Linkability

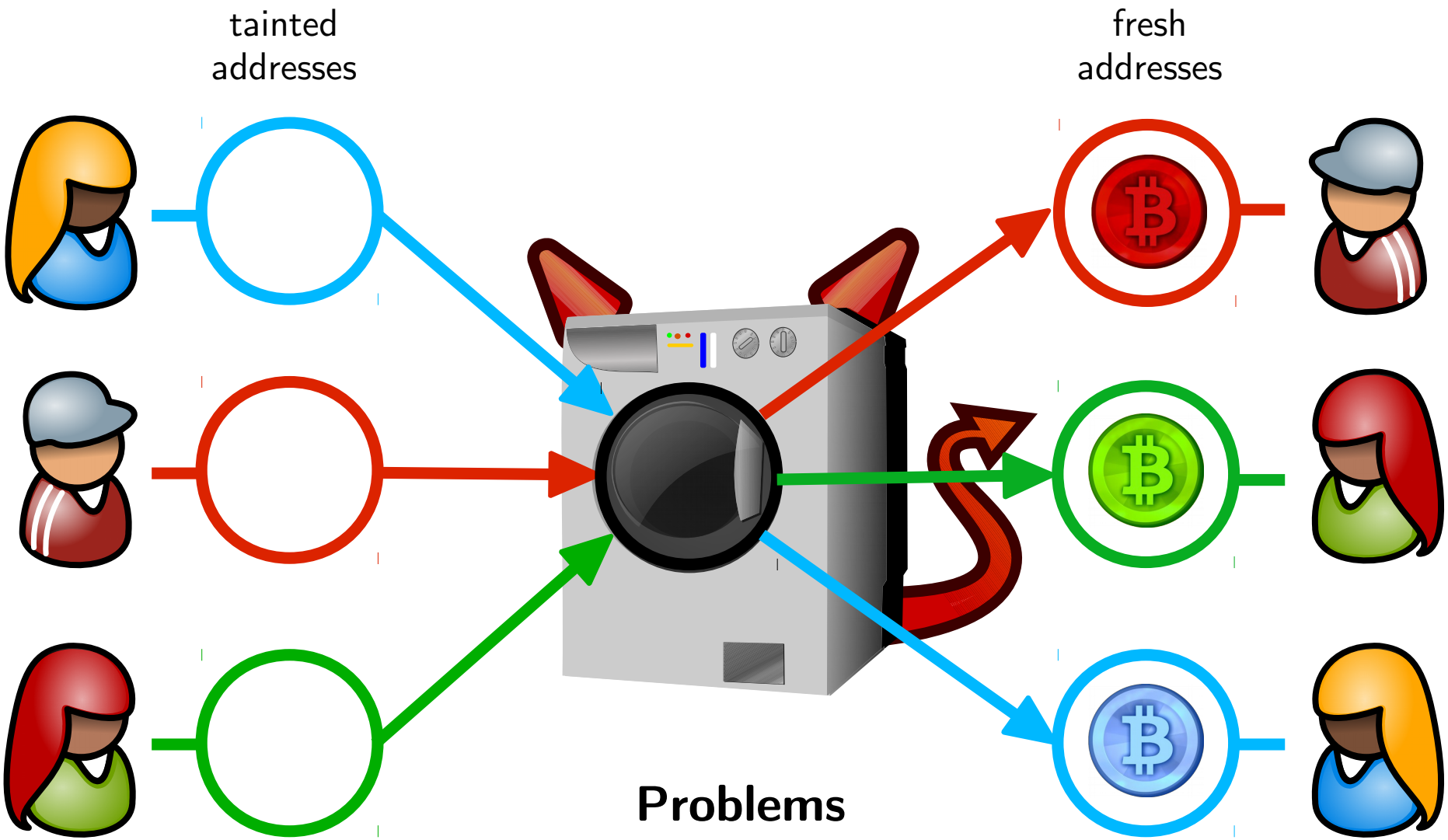
Zerocoin [Miers et al., S&P'13]
and **Zerocash** [Ben-Sasson et al., S&P'14]

- ✔ Unlinkability by design
- ✗ Inefficient or requires trusted setup
- ✗ Not compatible with Bitcoin

Naïve Coin Mixing



Naïve Coin Mixing

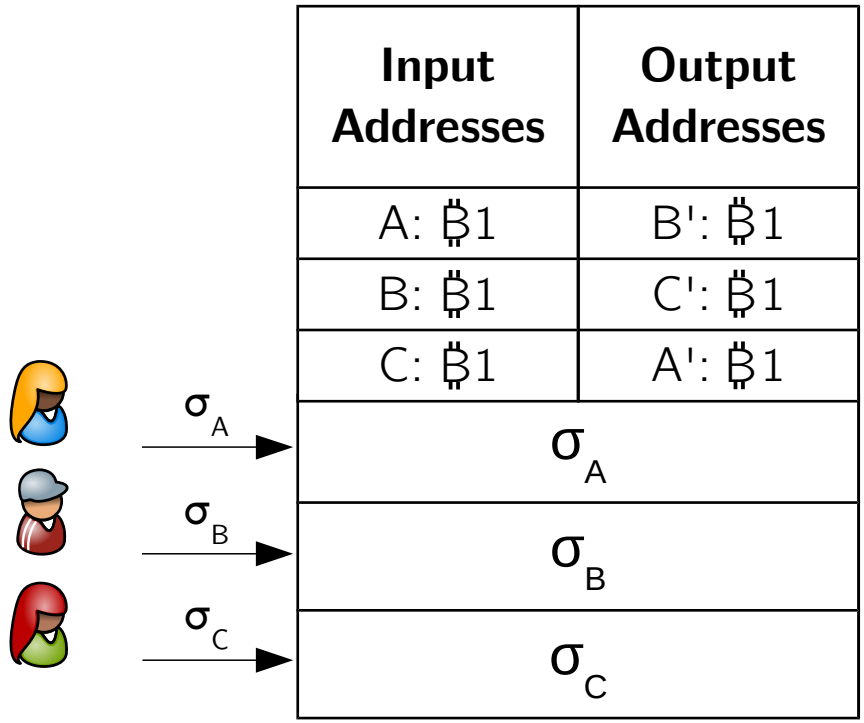


Problems

- Mix can still link the addresses
- Mix can steal the money
- Mix may require a fee

accountability possible using Mixcoin [Bonneau et al., FC'14]

Mixing using CoinJoin [Maxwell]



✔ Coins cannot be stolen. **(Verifiability)**

✗ How to create shuffled list of outputs obliviously? **(Unlinkability)**

✗ How to prevent DoS attacks? **(Robustness)**

Goals

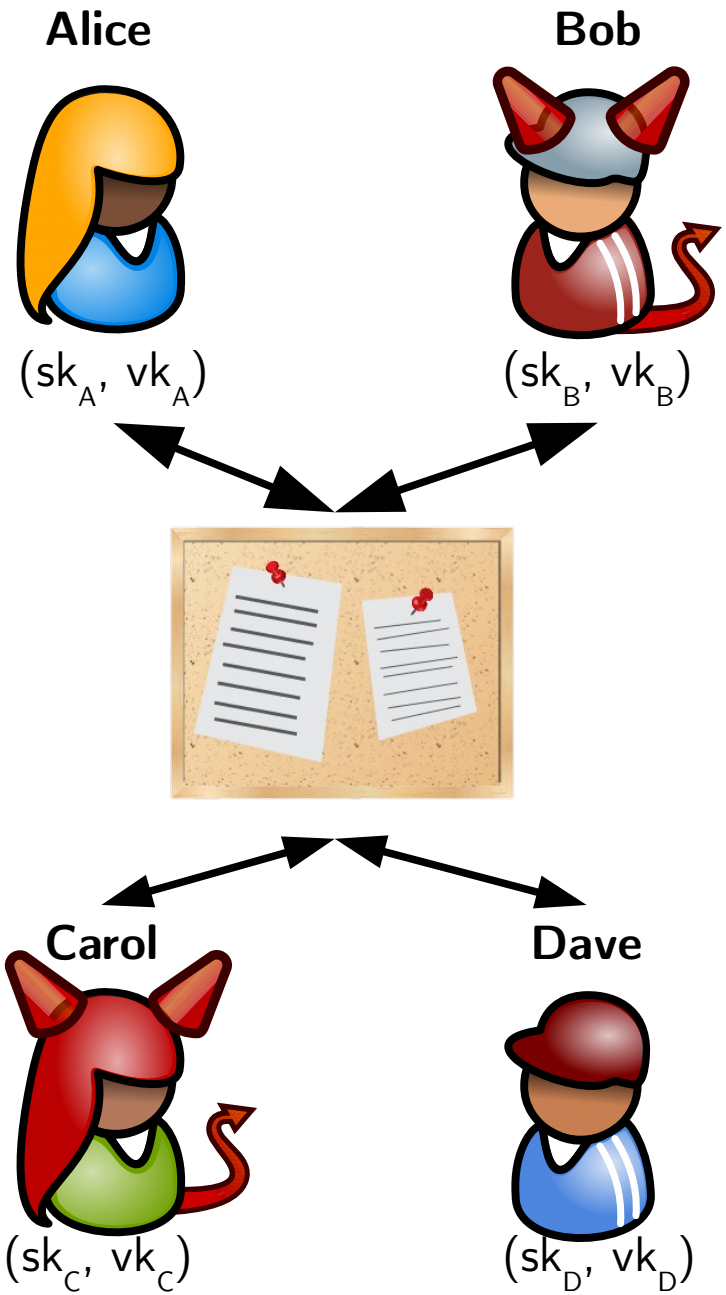
Security

- **Unlinkability**
  After a successful run of the protocol, participants' input and output addresses are unlinkable.
- **Verifiability**
  Coins cannot be stolen.
- **Robustness**
  If the protocol does not complete, we can expose at least one misbehaving user (and restart without this user).

System

- **Compatibility with Bitcoin**
- **Decentralization**
- **Efficiency**

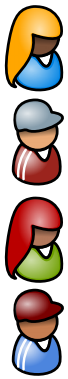
Scenario and Threat Model



- Bitcoin cryptographic keys
- Bulletin board does not exclude participants
- Two honest participants required for unlinkability

CoinShuffle: Announcement

ek: encryption key
dk: decryption key
sk: signing key



$A' \leftarrow \text{AddrGen}();$

$(ek_B, dk_B) \leftarrow \text{EncGen}();\ B' \leftarrow \text{AddrGen}();$

$(ek_C, dk_C) \leftarrow \text{EncGen}();\ C' \leftarrow \text{AddrGen}();$

$(ek_D, dk_D) \leftarrow \text{EncGen}();\ D' \leftarrow \text{AddrGen}();$

$\text{Sign}(sk_A; A)$

$\text{Sign}(sk_B; ek_B, B)$

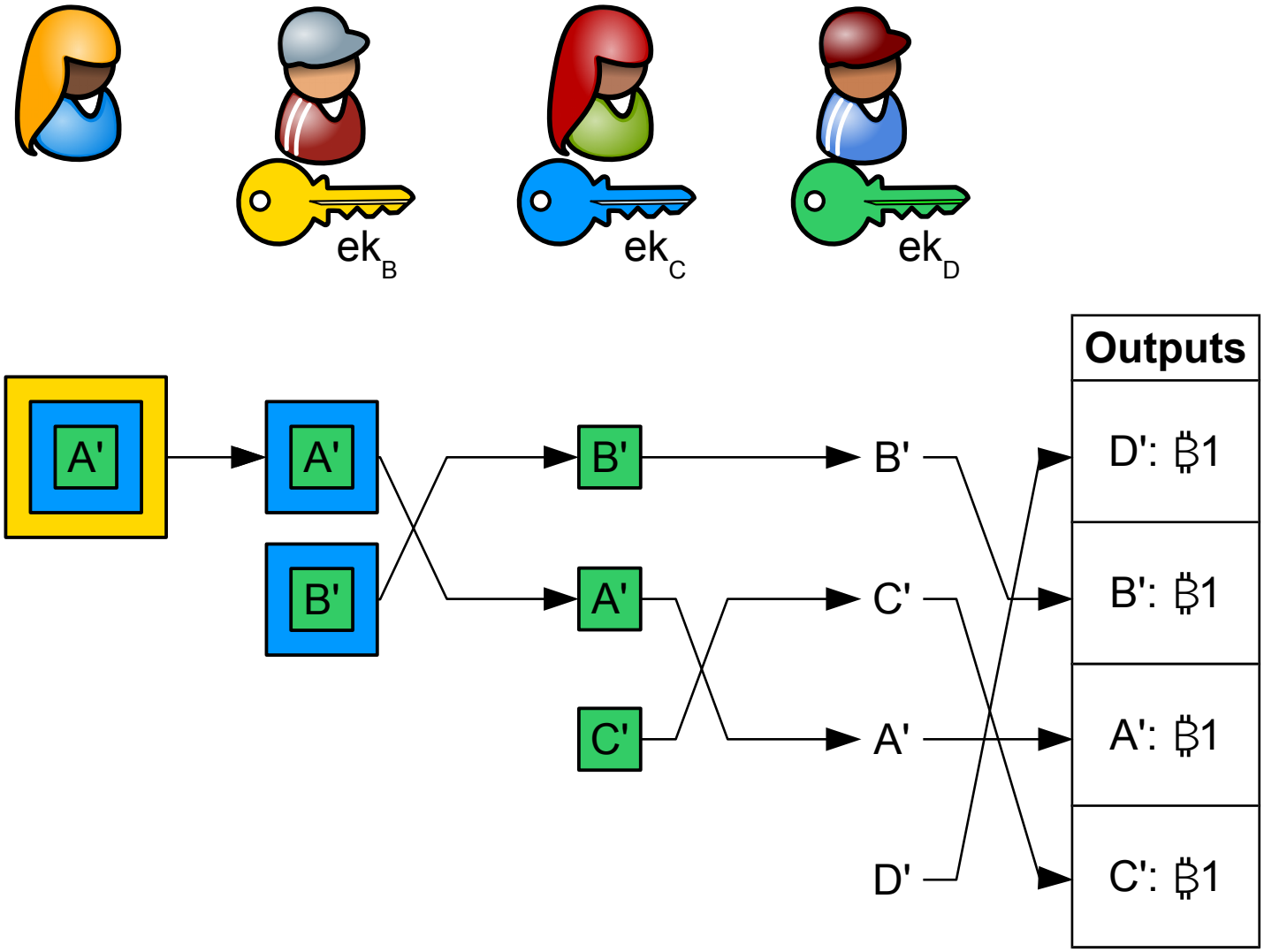
$\text{Sign}(sk_C; ek_C, C)$

$\text{Sign}(sk_D; ek_D, D)$

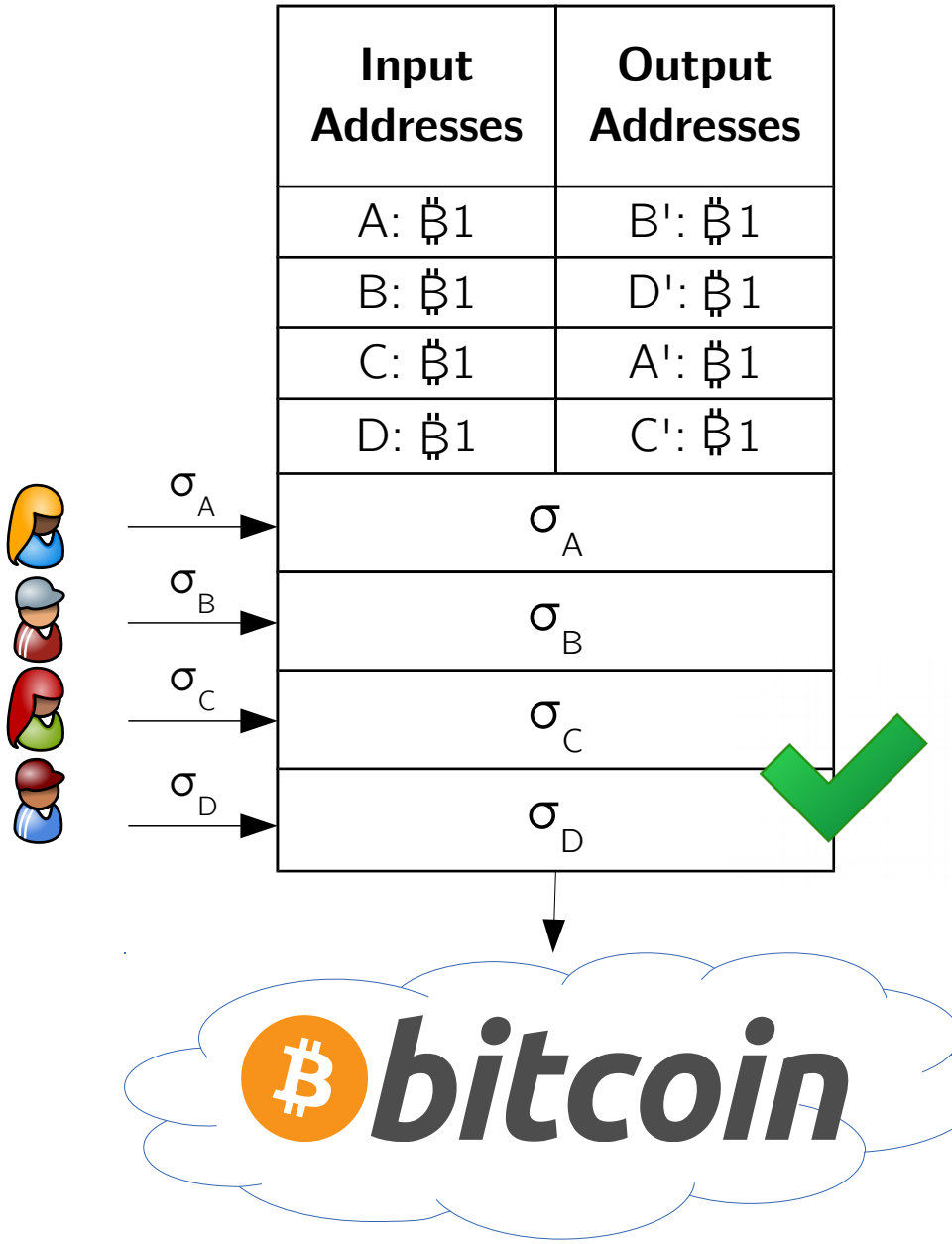
Input Addresses
A: ฿1
B: ฿1
C: ฿1
D: ฿1

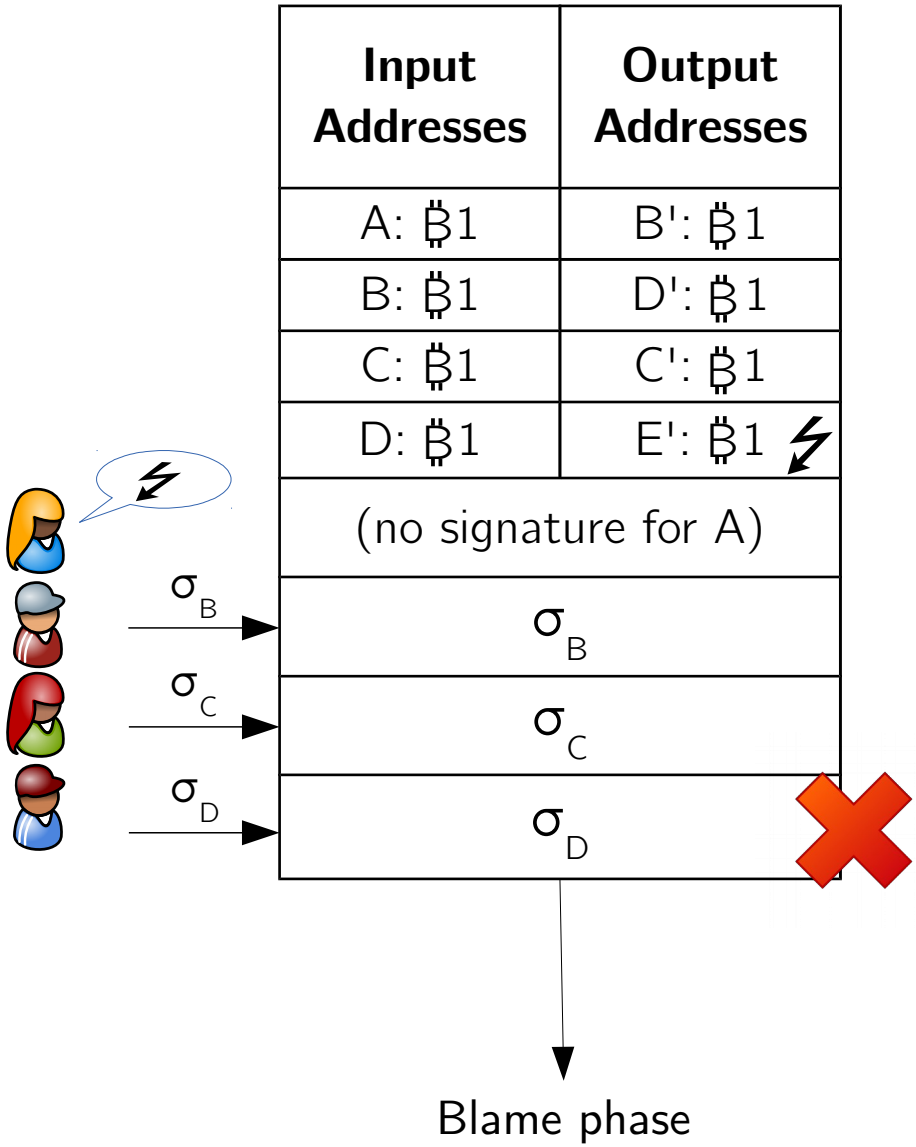
CoinShuffle: Shuffling

inspired from Dissent [Corrigan-Gibbs and Ford, CCS'10], with 4x speedup

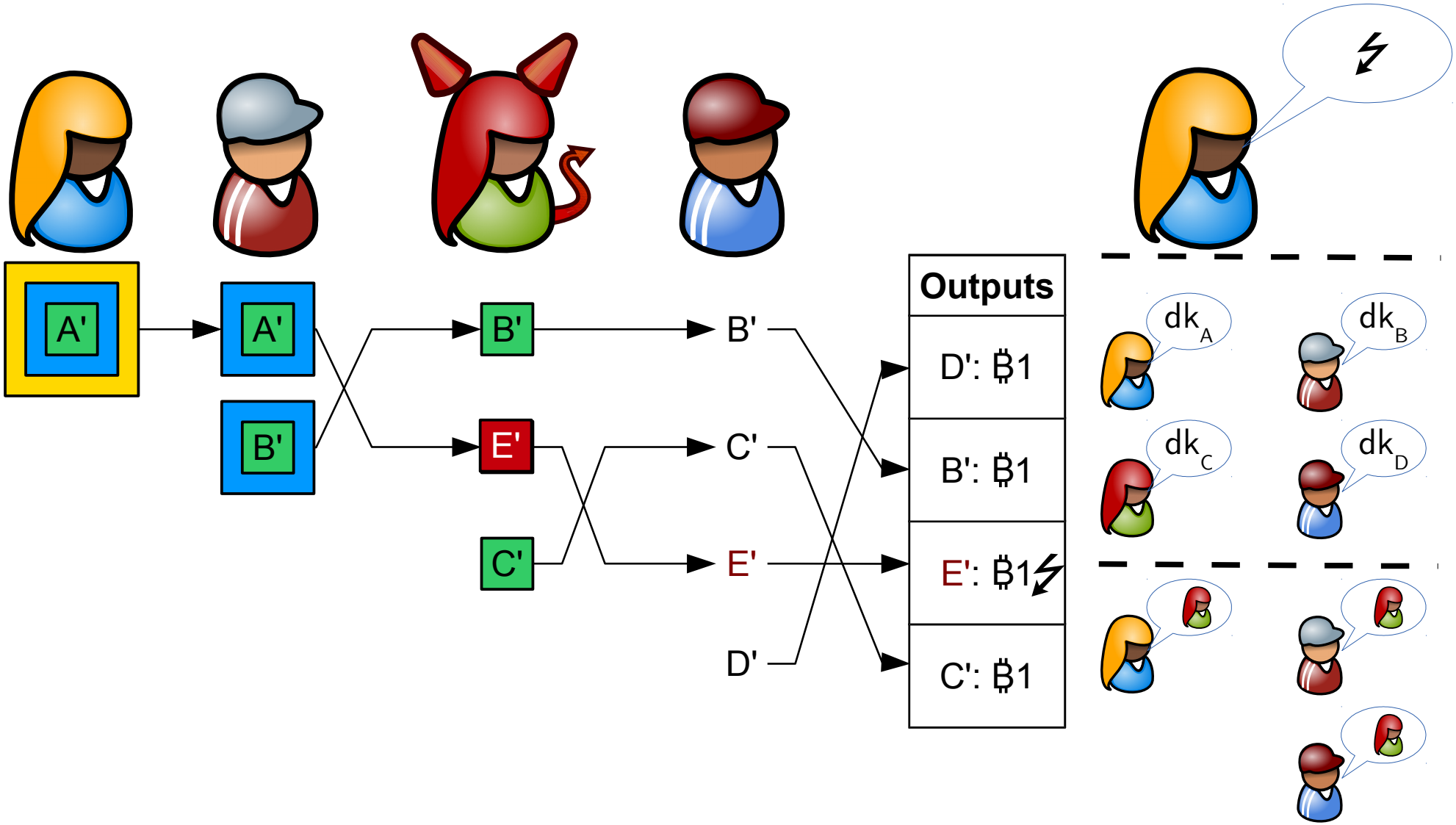


CoinShuffle: Transaction Verification





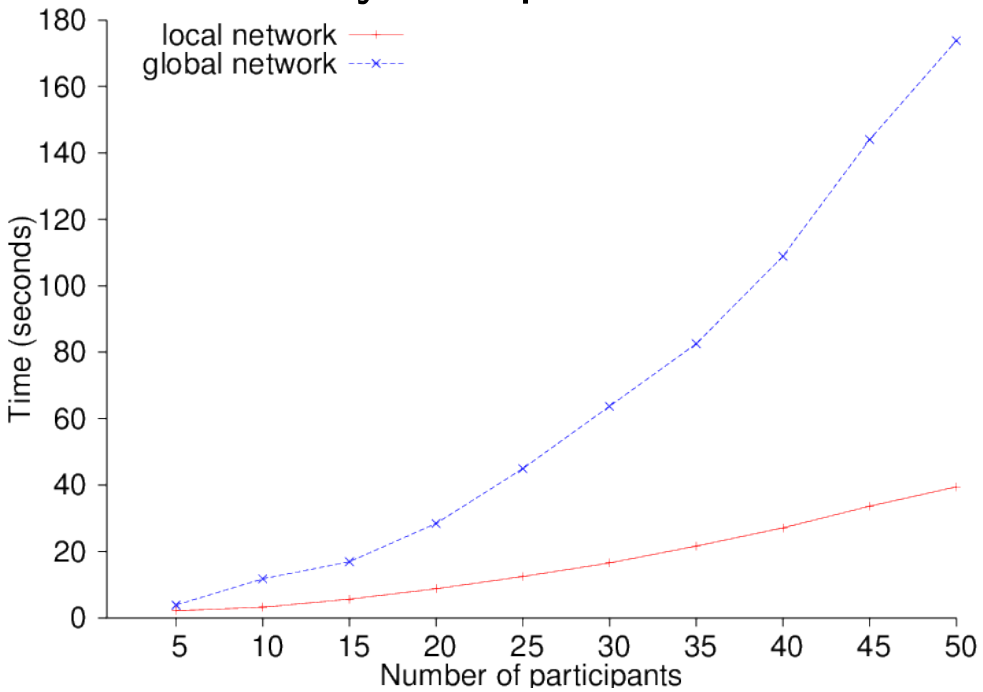
CoinShuffle: Blame

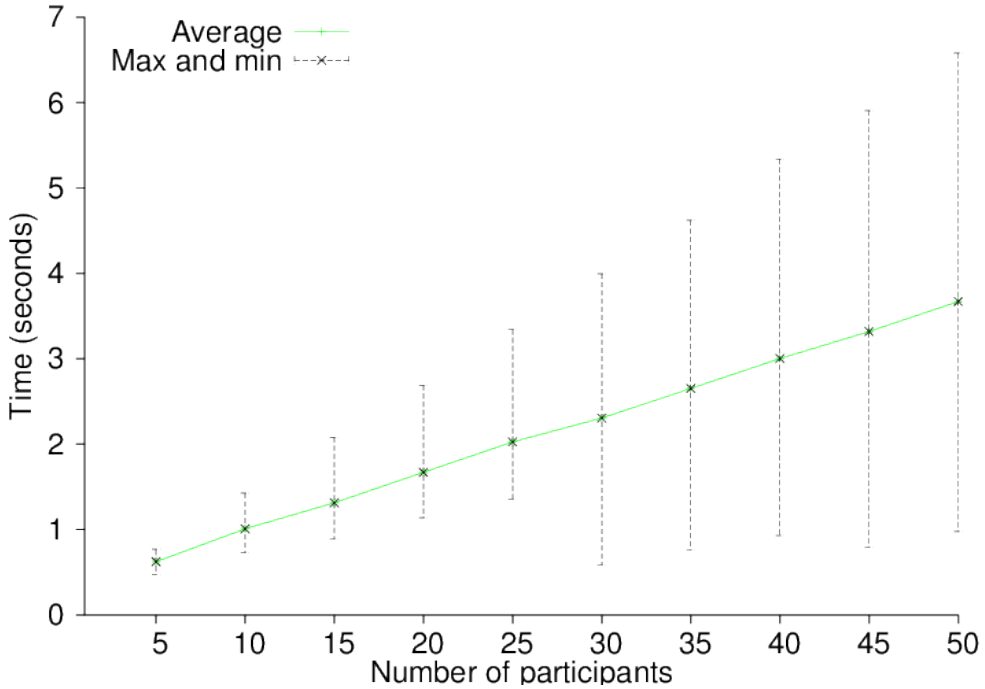


All (other) cases are discussed in the paper in detail.

Evaluation

- Prototype implementation in Python
- OpenSSL cryptographic library
- Emulab testbed
- Measured time for:
  - Running the whole protocol (local and global network scenarios)
  - Only computation





Communication with the Bitcoin Community

Demo (Bryan Vu)

Bitcoin SF Devs Seminar: A deep dive into Coinshuffle & Decentralized coinjoin

August 11 · 7:00 PM
20Mission

Join Bryan Vu (Google) as he shares his work on a deep dive into a new coin mixing protocol to improve anonymity in Bitcoin. The protocol is called CoinShuffle.

meetup.com/SF-Bitcoin-Devs/events/195004712/

BitcoinAuthenticator (Chris Pacia)

youtube.com/watch?v=T-CX-S8fq5A

NXT

Come-from-Beyond
@comefrombeyond

Following

Monetary System is delayed because I'm adding #CoinShuffle to it. Yes, you read it correctly, #Nxt will have anonymous payments.

Reply Retweeted Favorited More

RETWEETS 33 FAVORITES 12

4:25 PM - 5 Jul 2014

twitter.com/comefrombeyond/status/485429369268350977

Conclusions

- Linkability of transactions is a privacy concern for users
- We present CoinShuffle:
  - A decentralized, efficient and totally compatible protocol
  - Based on mixing and CoinJoin
  - Security guarantees: unlinkability, verifiability and robustness
- Next step: Specification and real implementation
- Project page with paper and prototype implementation: http://crypsys.mmci.uni-saarland.de/projects/CoinShuffle

Contact:
Tim Ruffing
tim.ruffing@mmci.uni-saarland.de
@real_or_random