Anonymity in Cryptocurrencies

Foteini Baldimtsi

10. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.

Satoshi Nakamoto, 2008

Bitcoin offers privacy—as long as you don't cash out or spend it

or the public ledger that records bit
bitcoins move from one person to a
alphanumeric addresses.

A Fistful of Bitcoins: Characterizing Payments Among Men with No Names

Sarah Meiklejohn Marjori Pomarole Grant Jordan
ill Levchenko Damon McCoy† Geoffrey M. Voelker Stefan Savage
University of California, San Diego George Mason University†

Quantitative Analysis of the Full Bitcoin Transaction Graph

Dorit Ron and Adi Shamir

Department of Computer Science and Applied Mathematics,
The Weizmann Institute of Science, Israel
{dorit.ron,adi.shamir}@weizmann.ac.il

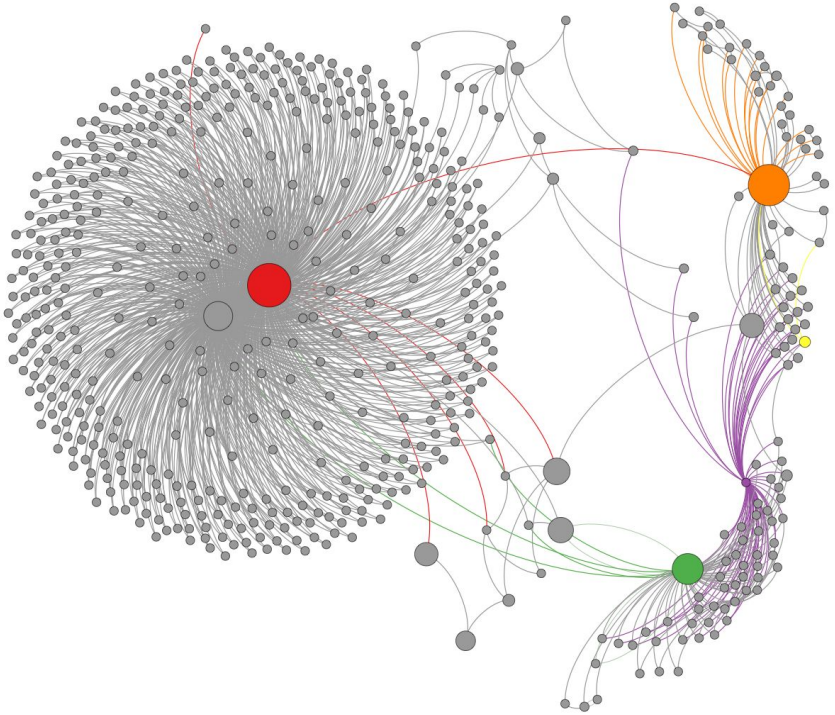
Evaluating User Privacy in Bitcoin

Elli Androulaki[1], Ghassan O. Karame[2], Marc Roeschlin[1],
Tobias Scherer[1], and Srdjan Capkun[1]

Bitcoin is only pseudonymous

If anyone is ever able to link your Bitcoin address to your real world identity, then all of your transactions — past, present, and future — will have been linked back to your identity.

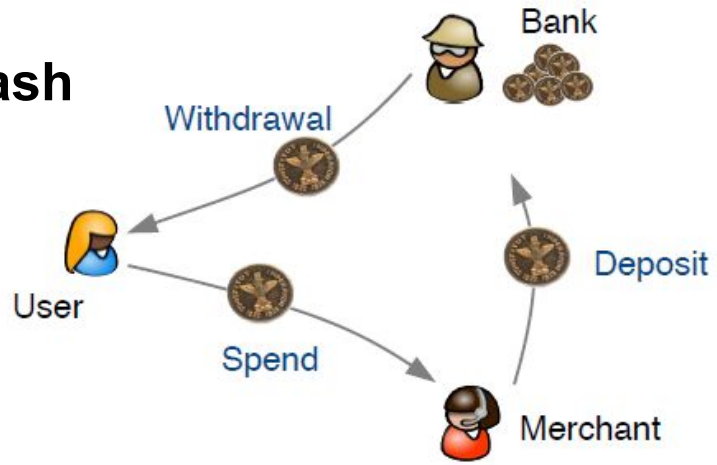
De-anonymizing Bitcoin users



Bitcoin De-anonymization in Practice

Anonymity: the goal

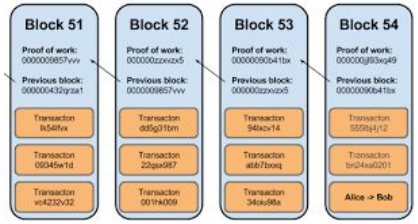
eCash



Adversarial Bank cannot link a withdrawal to a deposit

unlinkability

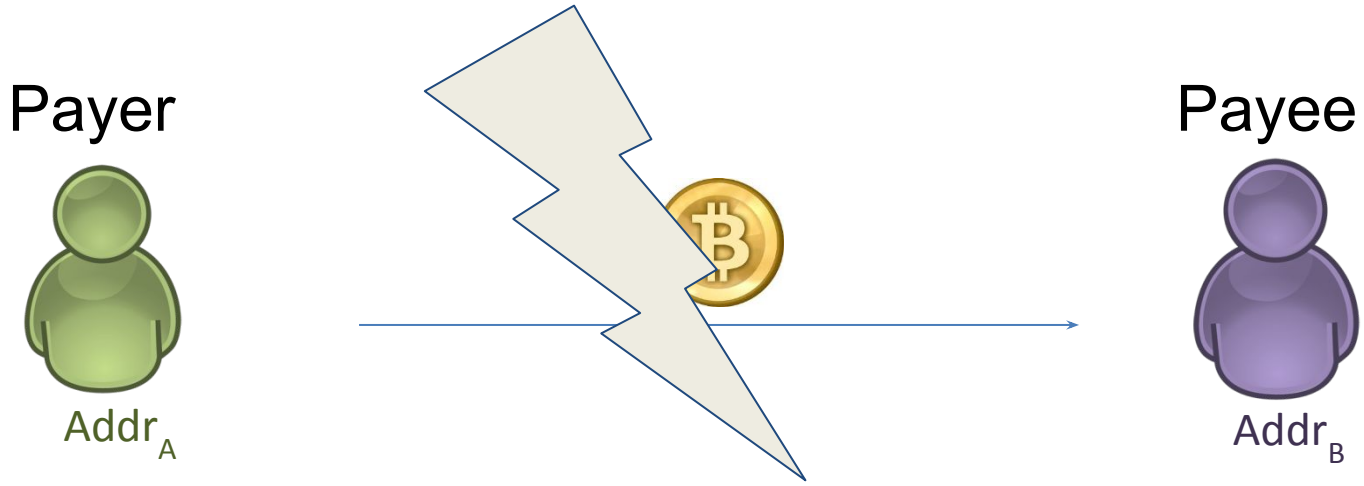
Bitcoin



Ledger

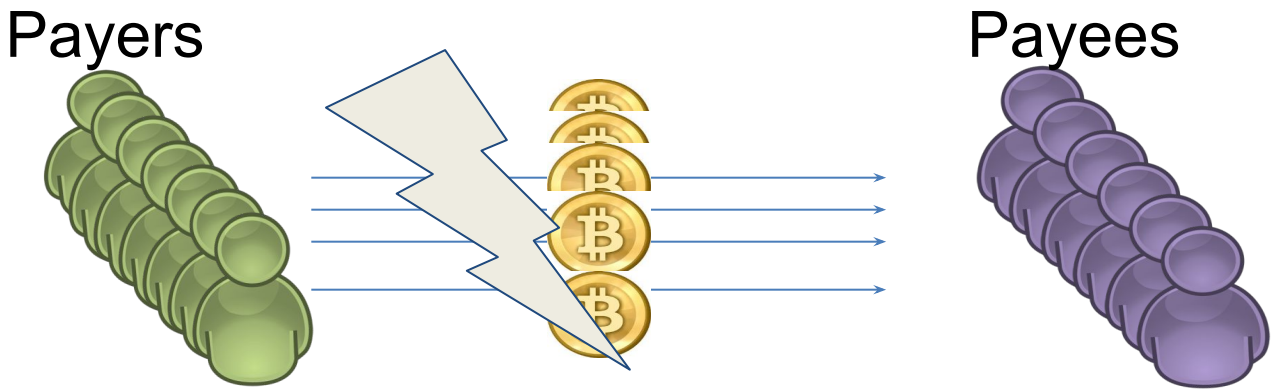
It should be hard to link the sender of a payment to its recipient

Anonymity: the goal



Break the link between payer and payee

Anonymity Flavors



Set Anonymity: the set of transactions which the adversary cannot distinguish from your transaction (depends on anonymity model)

Two Main Directions

1) Mixing/Tumbler Services (for Bitcoin)

CoinShuffle

mixcoin
True Anonymous Cryptocurrency

CoinSwap
Simple. Fast. Exchange

Blindcoin

CoinJoin:

XIM

Bitcoin Compatible

2) Anonymous Cryptocurrencies

ZCASH

Zerocoin

MONERO

Non-Compatible to Bitcoin

Why do we need anonymity

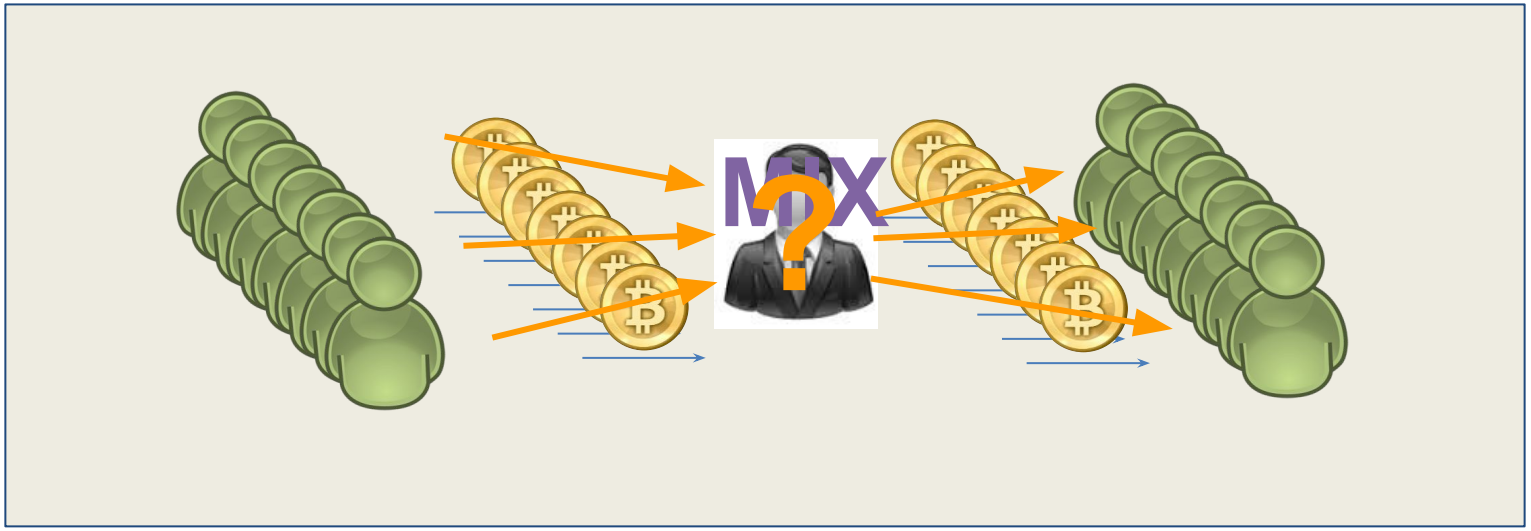
- achieve the level of privacy that we are already used to from traditional banking, and mitigate the deanonymization risk that the public blockchain brings.

- go above and beyond the privacy level of traditional banking and develop currencies that make it technologically infeasible for anyone to track the participants.

PART I

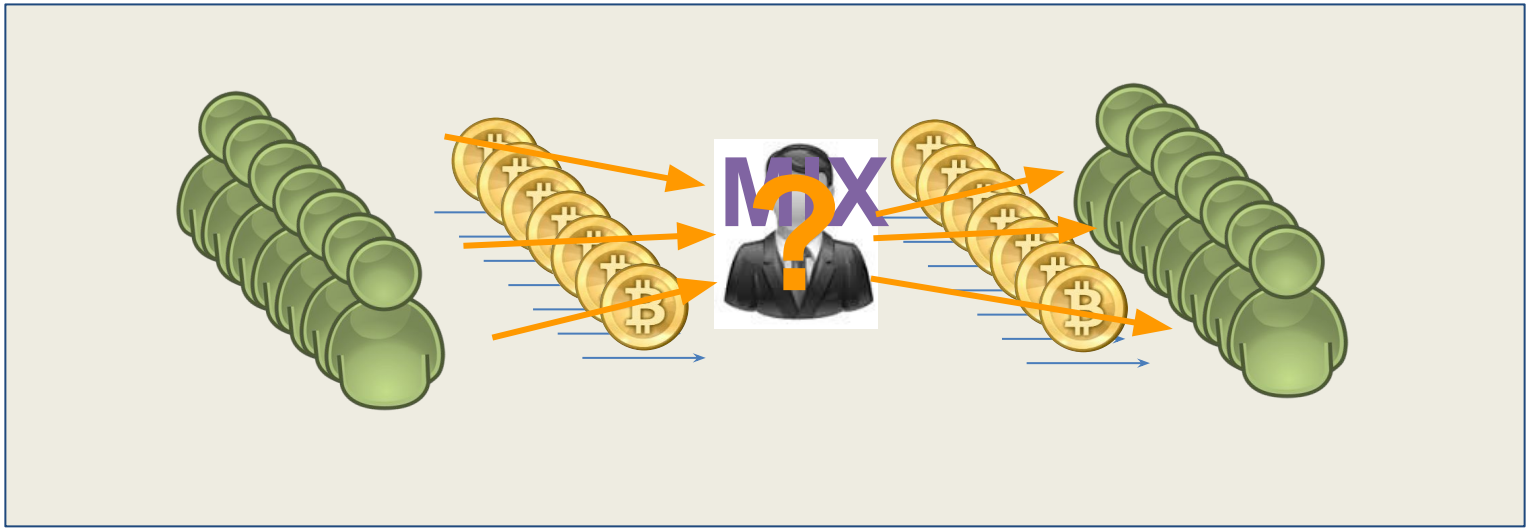
Mixing/Tumbler Services

What is a mix?



- Centralized (intermediary)
- Decentralized (i.e. Coinshuffle)

What is a mix?



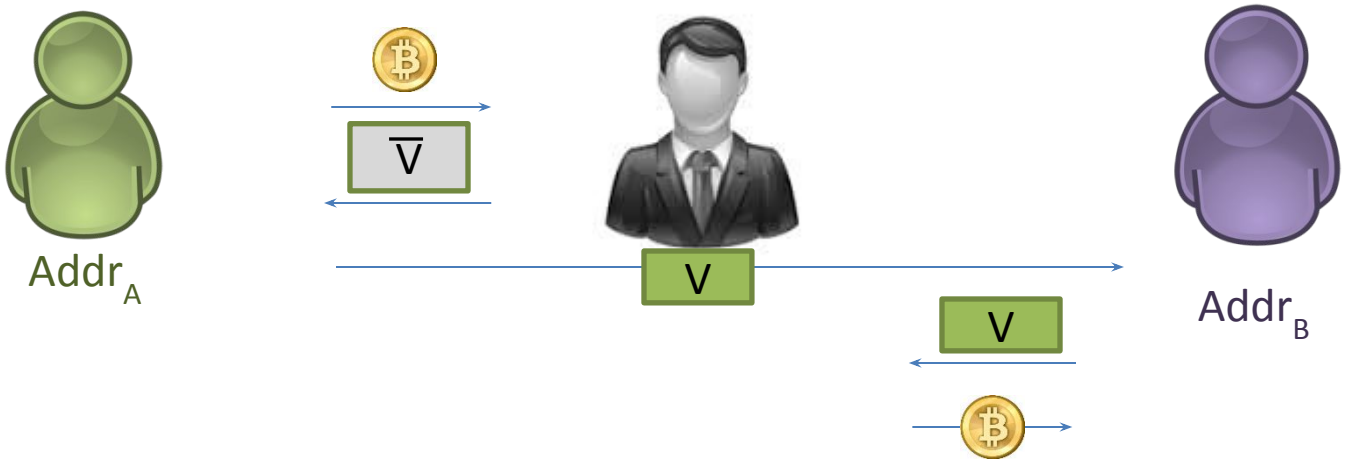
2 challenges

- privacy against intermediary
- security against intermediary

Attempt 1 - Centralized Scheme

Intermediary blindly issues vouchers?

Goal: Set-Anonymity

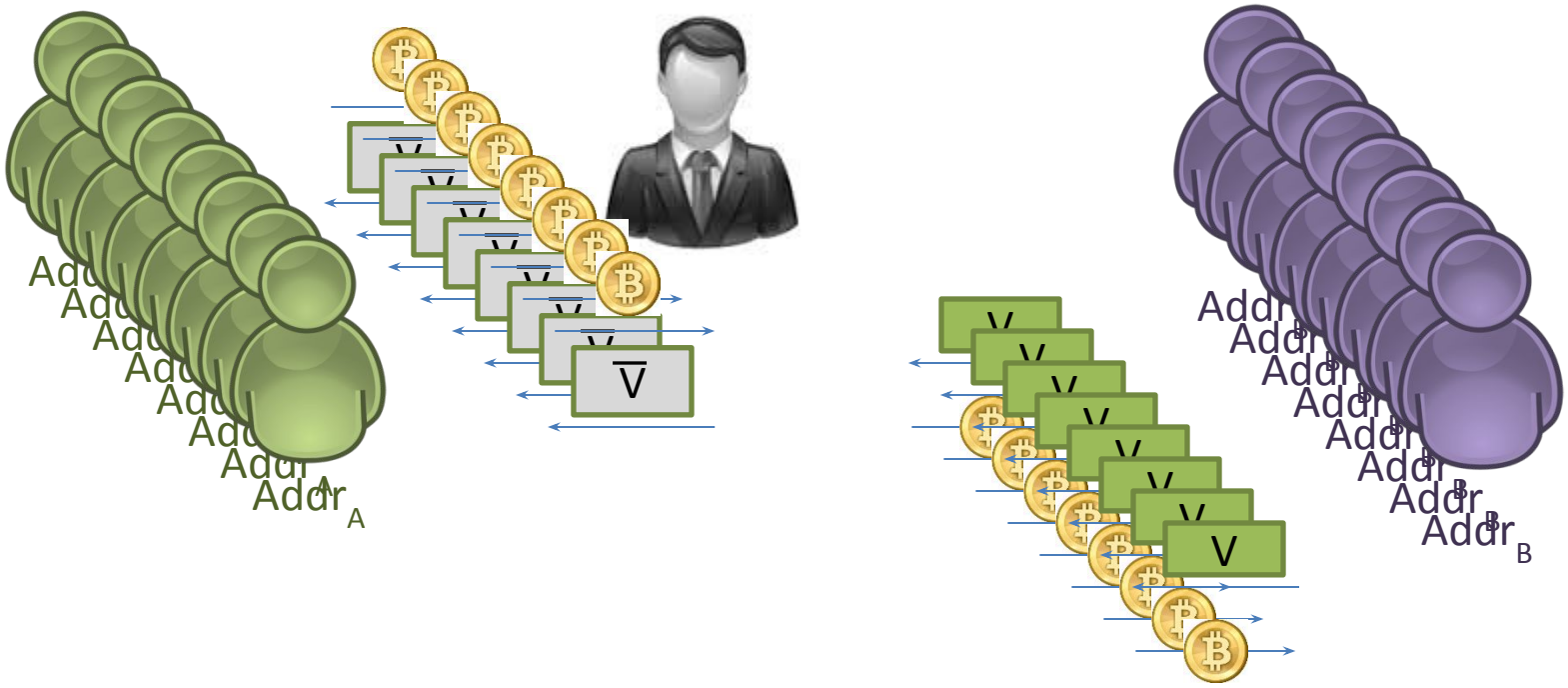


Intermediary cannot link a voucher it issued to a voucher it redeems!

- **Blind signatures**

Intermediary blindly issues vouchers?

Goal: **Set-Anonymity**

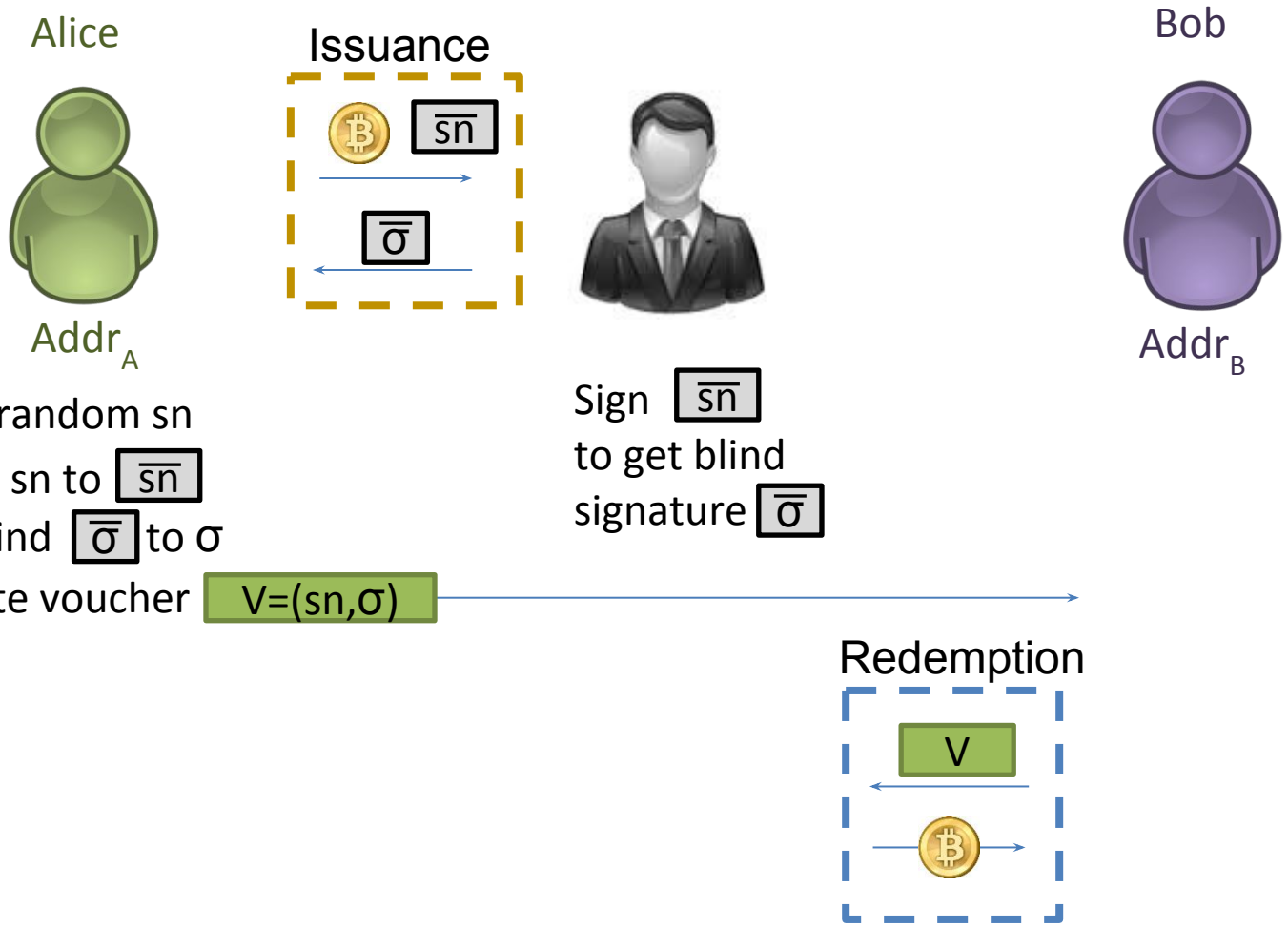


Intermediary cannot link a voucher it issued to a voucher it redeems!

- **Blind signatures**

Attempt 2 - Centralized Scheme

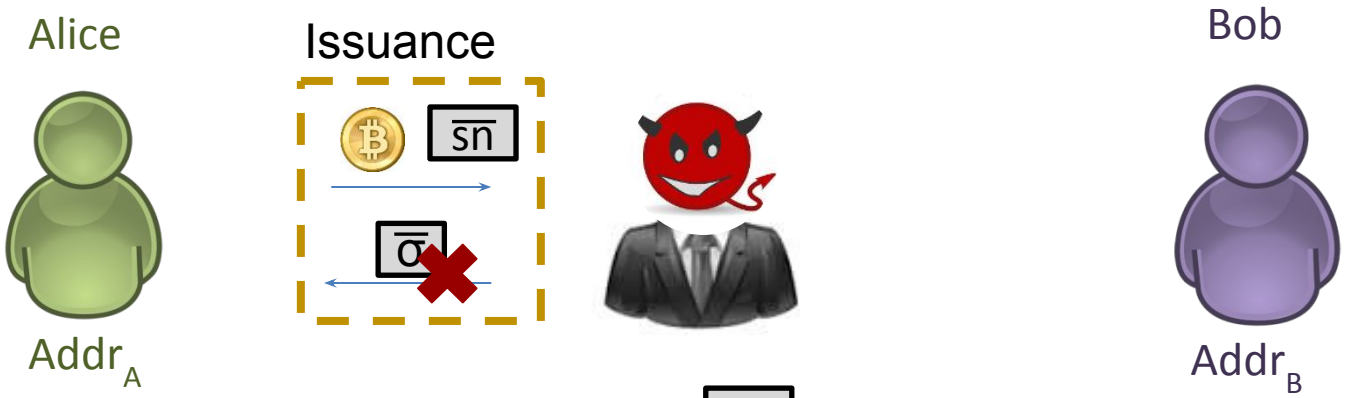
Intermediary blindly issues vouchers?



Attempt 2 - Centralized Scheme

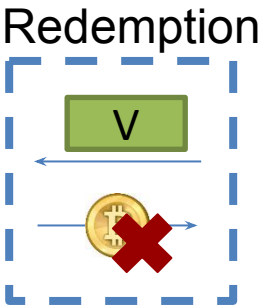
Intermediary blindly issues vouchers?

But what if Intermediary is malicious and refuses to issue $\overline{\sigma}$ or return ₿ ?



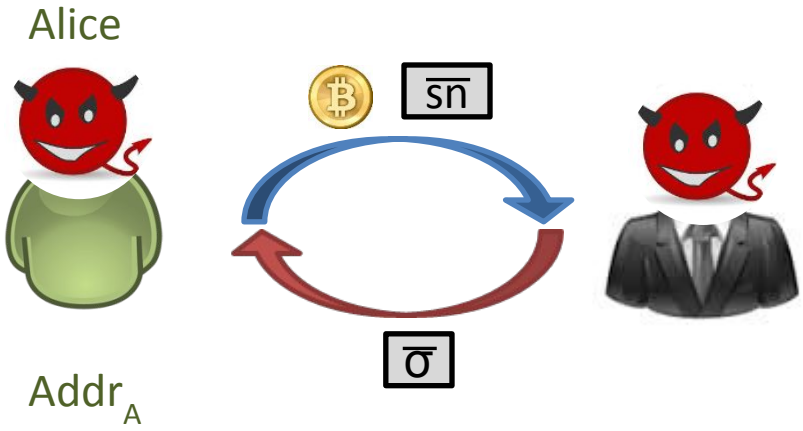
1. Pick random sn
2. Blind sn to $\overline{sn}$
3. Unblind $\overline{\sigma}$ to σ
4. Create voucher V=(sn,σ)

Sign $\overline{sn}$ to get blind signature $\overline{\sigma}$



Blindly Signed Transaction Contracts

Goal: Set-Anonymity, **Fair Exchange/Atomic swaps**



Transaction Offer: V for ₿.

"Addr$_A$ pays ₿ to a spending transaction that has a valid blind signature on sn. This must be done within time tw."

Transaction Fulfill: V for ₿.

"Here is σ."

Fair exchange is robust if either party is malicious!

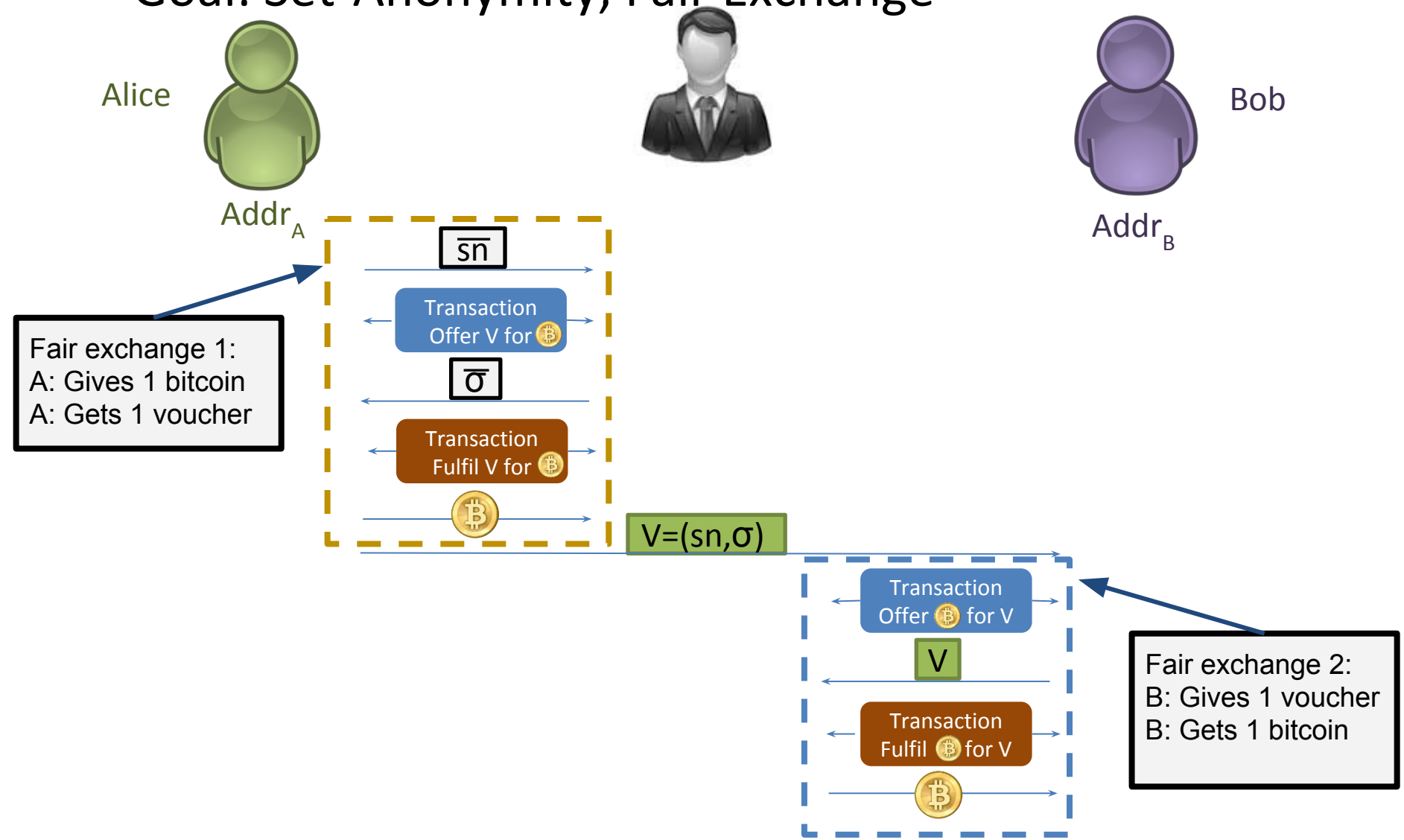
- **Bitcoin Scripts***

*** The blind signature we use requires a soft fork**

Attempt 3 - centralized scheme

Blindly Signed Transaction Contracts

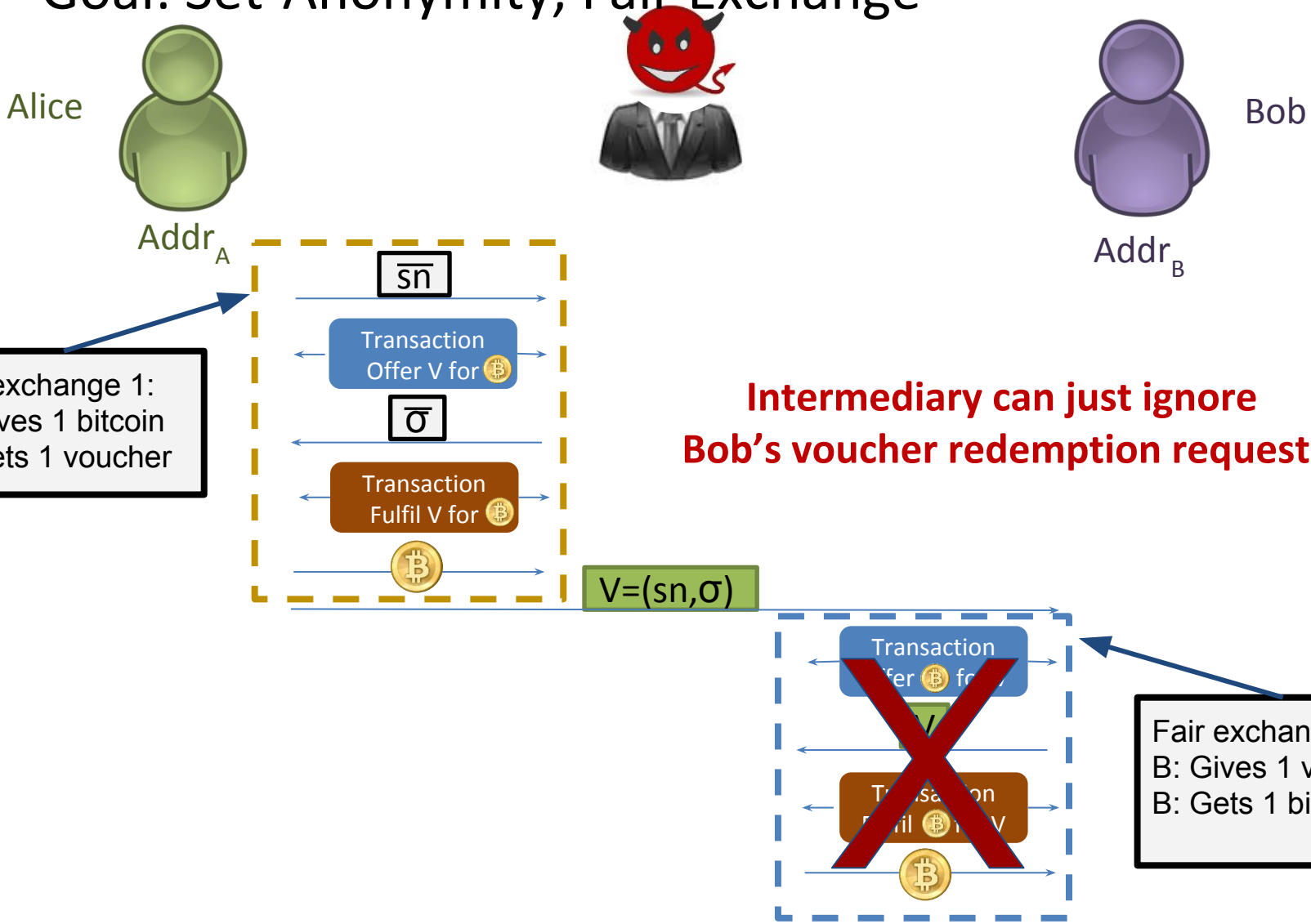
Goal: Set-Anonymity, Fair Exchange



Attempt 3 - centralized scheme

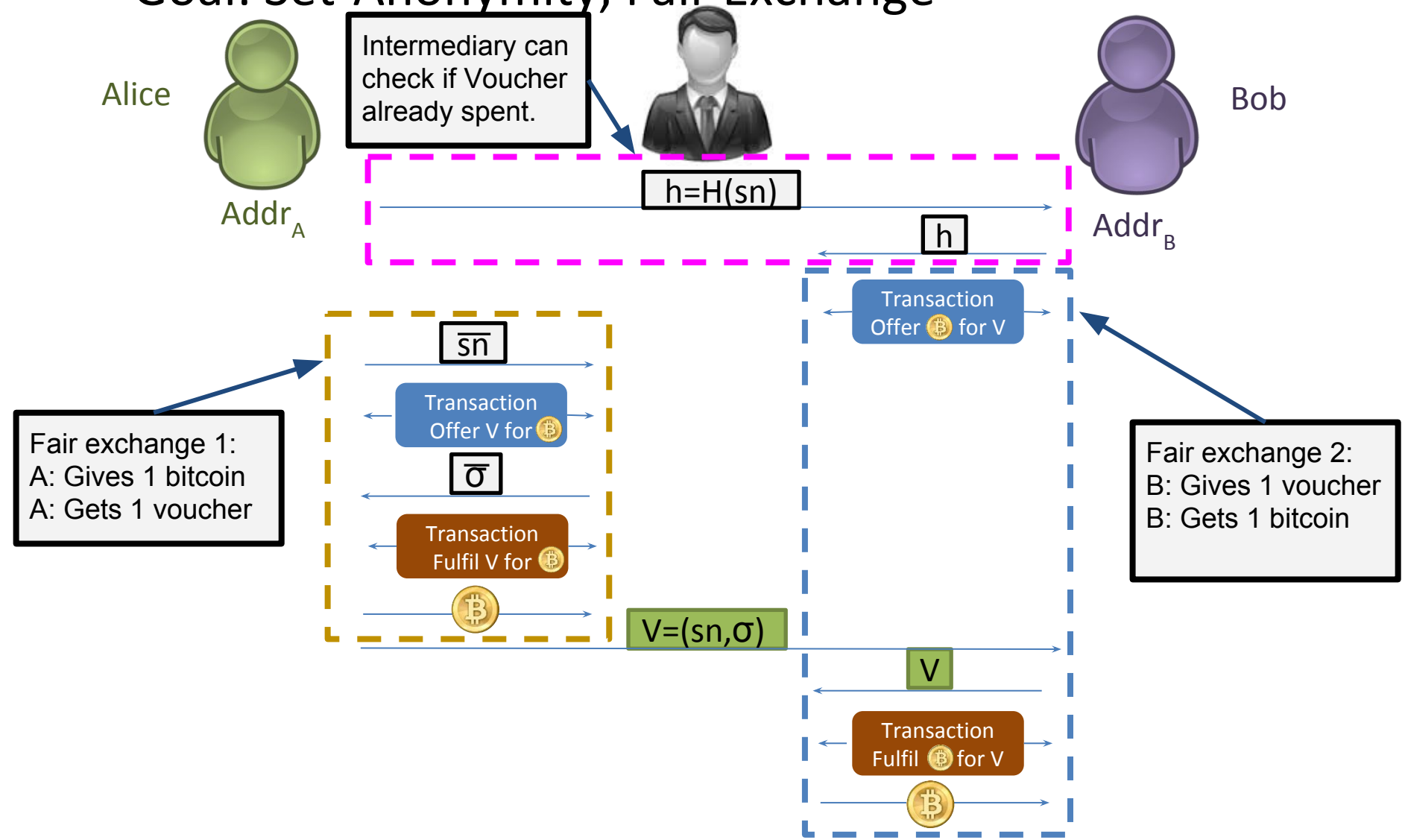
Blindly Signed Transaction Contracts

Goal: Set-Anonymity, Fair Exchange



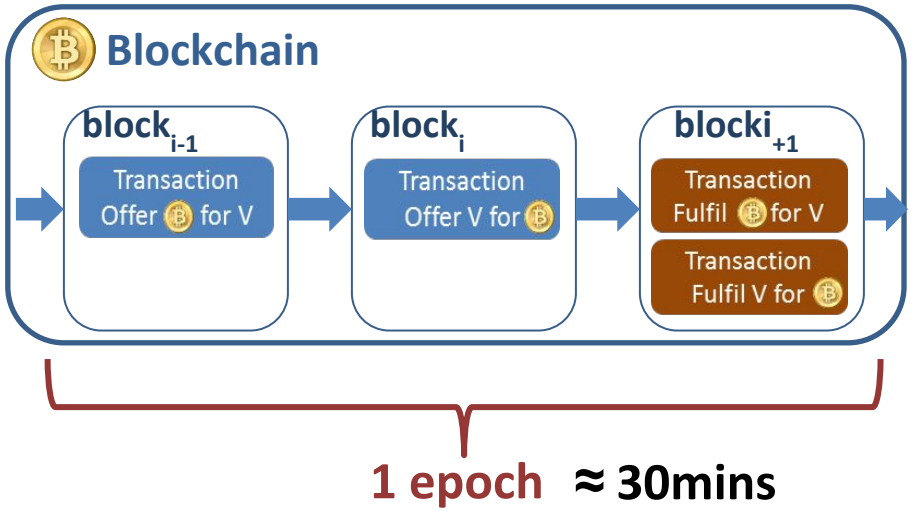
Blindly Signed Transaction Contracts

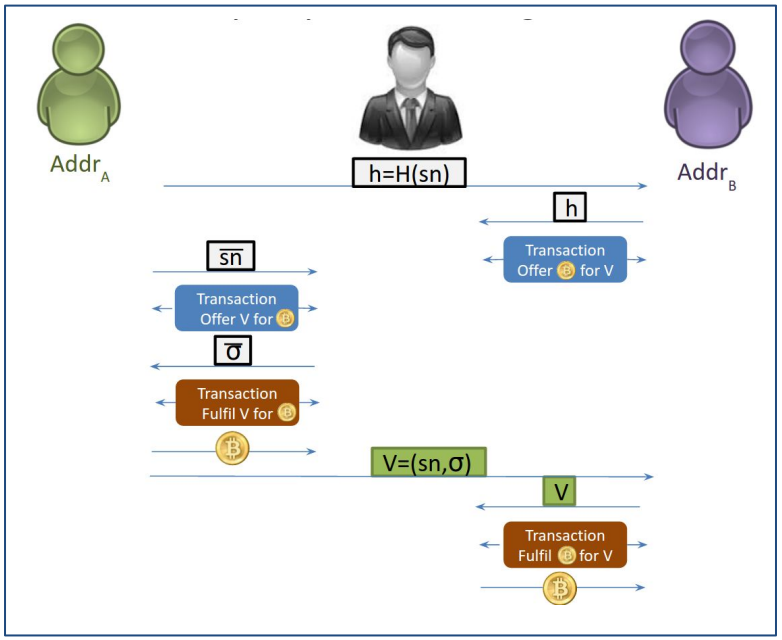
Goal: Set-Anonymity, Fair Exchange



Blindly Signed Transaction Contracts

What is stored on the blockchain?





Anonymity properties:

1. **Set Anonymity within an Epoch.**
2. **Transparency of Anonymity Set.**

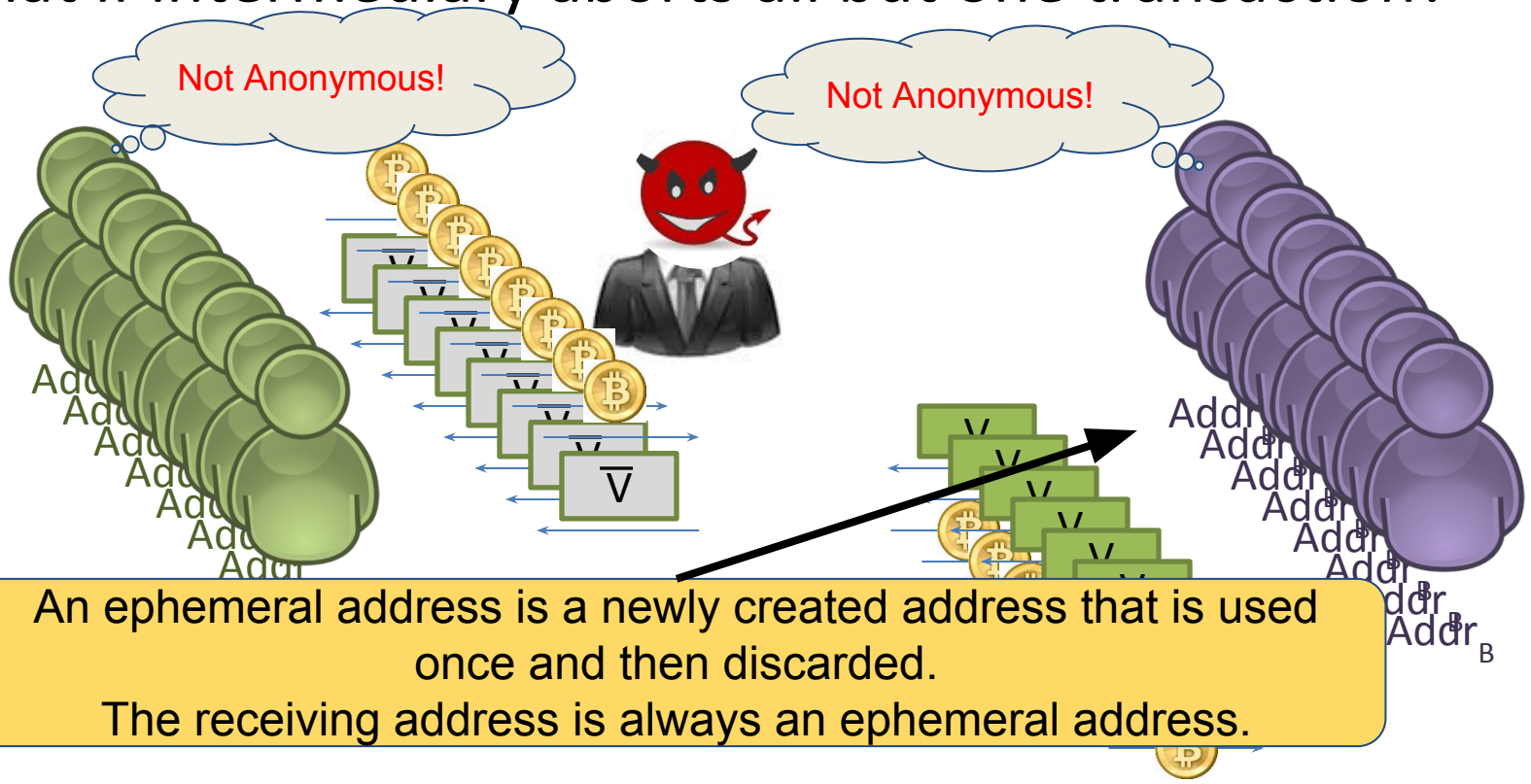
How do we achieve this?

(resists a fully malicious intermediary!)

(It's visible on the blockchain)

Anonymity vs Malicious Intermediary?

What if intermediary aborts all but one transaction?



Countermeasures:

1. **Small anonymity set is visible on the blockchain.**
2. **$Addr_B$ is ephemeral; If anonymity set is too small anonymously send it a new ephemeral addr (rinse & repeat).**

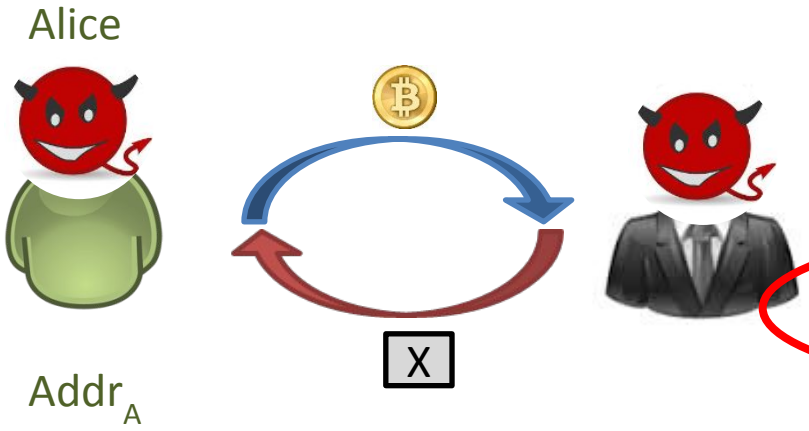
Anonymity vs Malicious Intermediary?

What if intermediary distort anonymity set transparency with sybils?

- **Expensive due to sybil resistance:**
  - Intermediary pays all transaction fees for each sybil.
- **Low success rate:**
  - If intermediary waits until it sees Alice's address to abort, Alice and Bob can detect attack.
  - If intermediary launches the attack earlier, it only sees Bob's address which is an ephemeral address (untargeted).

Background: Bitcoin Transaction Contracts

Goal: **Fair Exchange/Atomic swaps:**



Transaction Offer: X for ฿.

"Addr$_A$ pays ฿ to a spending transaction has a value X satisfying condition C.

Transaction Fulfill: X for ฿.

"Here is X ."

Bitcoin transaction scripts are very limited.
We can only check two types of cryptographic conditions C:

1. Hash(X) = Y,
2. ECDSA_CheckSignature(Tx, PUBLIC_KEY) = TRUE

Big Picture

New Cryptocurrencies

Not compatible with bitcoin

HBG'16

TumbleBit 

Vulnerable to DoS & Sybil Attacks

Bitcoin-Compatible Schemes

(aka "Mixing Services")

Vulnerable to bitcoin theft

Intermediary breaks anonymity

Mixing takes hours

Xim

PART II

Anonymous Decentralized Cryptocurrencies

Anonymous Decentralized Cryptocurrencies

Zerocoin: Anonymous Distributed E-Cash from Bitcoin

Ian Miers, Christina Garman, Matthew Green, Aviel D. Rubin
The Johns Hopkins University Department of Computer Science, Baltimore, USA
{*imiers, cgarman, mgreen, rubin*}*@cs.jhu.edu*

Almost a decentralized mixing service

performance issues and limited functionality

Zerocash: Decentralized Anonymous Payments from Bitcoin
(extended version)

Eli Ben-Sasson* Alessandro Chiesa† Christina Garman‡ Matthew Green‡
Ian Miers‡ Eran Tromer§ Madars Virza†

Standalone cryptocurrency

Zerocoin - main idea

Requires a trusted, append only bulletin board (it could be the Bitcoin blockchain)

Minting
pick SN, compute C1 = Commit(SN,r)
pin C1 on BB with a bitcoin

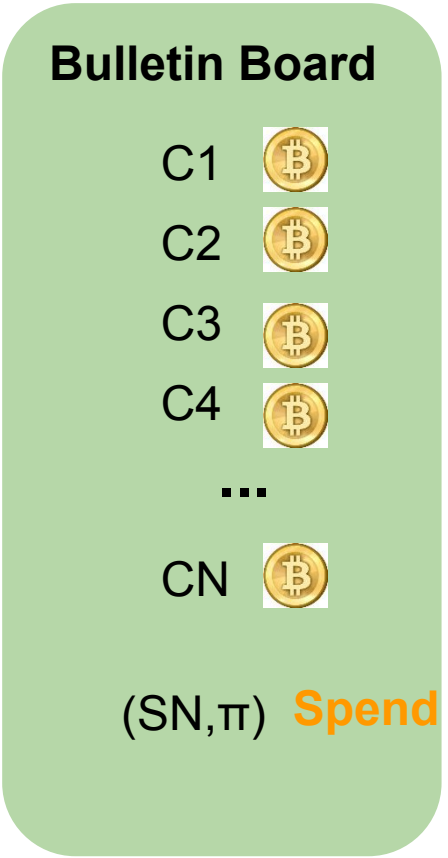
All Users accept C1 and agree it carries 1 ₿

Redeem
compute a NIZK π:
- I know Ci in (C1,C2,..,CN)
- I know r to open Ci to SN

Post (SN,π)

All Users verify π and check SN is new if OK, I can collect a ₿ from **any** location of BB

unlinkable by Commitment and NIZK



How to compute the proof π

Redeem
compute a NIZK π:
- I know Ci in (C1,C2,..,CN)
- I know r to open Ci to SN

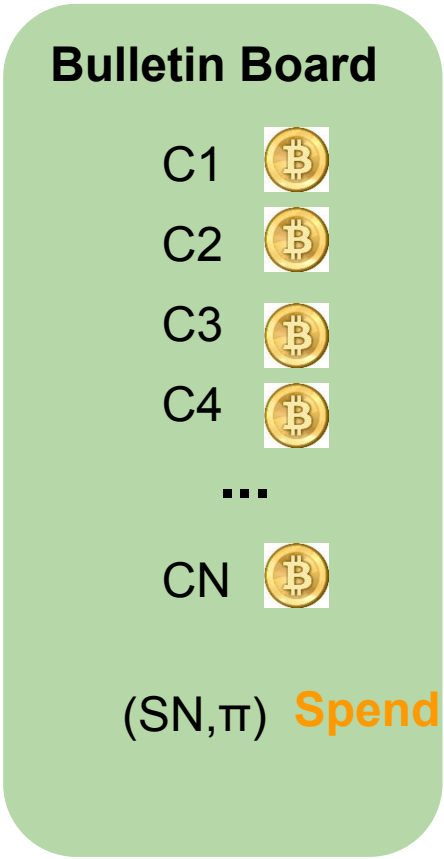
Post (SN,π)

Naive Solution

Identify all valid zerocoins in the bulletin board

Prove that SN is the serial number of a coin C
C = C1 ∨ C = C2 ∨ ...C=CN

This “OR” proof is O(N)



How to compute the proof π

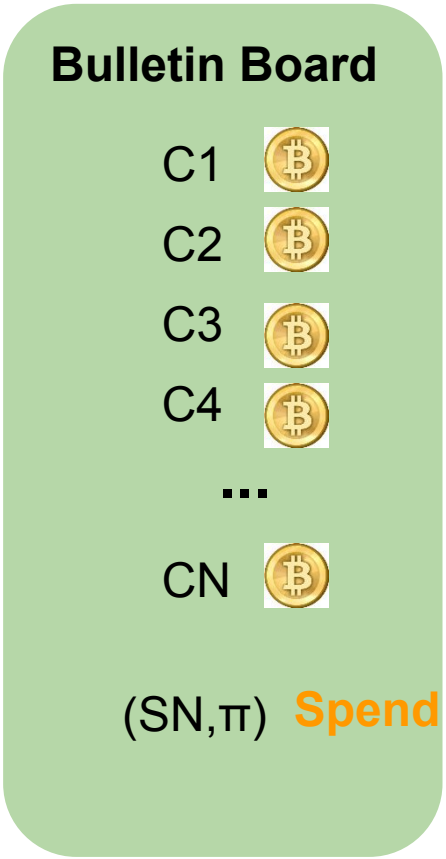
Cryptographic Accumulators

RSA modulus $n = p \cdot q$, $u \in QR_N$

Accumulator: $A = u^{C1\ C2\ ...CN} \bmod n$
witness for C2: $w = u^{C1\ C3\ ...CN} \bmod n$

To prove that C2 is in A give (w,C2)
check: $w^{C2} = A \bmod n$

This is not anonymous!



How to compute the proof π

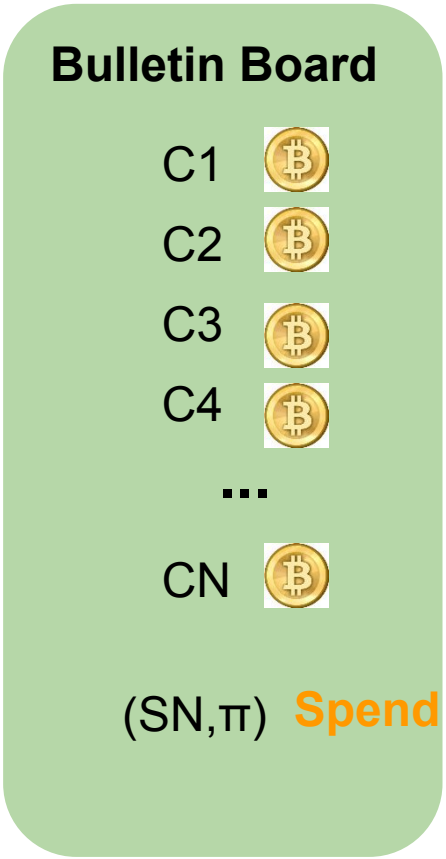
Cryptographic Accumulators

RSA modulus $n = p \cdot q$, $u \in QR_N$

Accumulator: $A = u^{C1\ C2\ ...CN} \bmod n$
witness for C2: $w = u^{C1\ C3\ ...CN} \bmod n$

To prove that C2 is in A give (w,C2)
check: $w^{C2} = A \bmod n$

There exists an efficient proof (NIZK) that I have a valid witness to a commitment of SN and know the corresponding randomness r
[CL'02] cost log (N)



Problems with Zerocoin

- Accumulators require **a trusted setup** (somebody to compute N and throw away p,q)
- Proofs **not very efficient** log(N)
  Each proof is approximately 50 KB) - note the scaling problems of Bitcoin
- **Not compatible** with bitcoin - these new types of transactions should be included - you would need to be able to verify sophisticated ZK proofs
- Payments of **single denomination and** payment values appear in **the clear** (1 BTC)

Zerocash **Solves the problems above***

Zerocash

Zerocash enables users to pay one another directly via payment transactions of variable denomination that reveal neither the origin, destination, or amount.

- reduces the size of transactions spending a coin to under 1 kB (an improvement of over 97:7%)
- reduces the spend-transaction verification time to under 6 ms (an improvement of over 98:6%)
- allows for anonymous transactions of variable amounts
- hides transaction amounts and the values of coins held by users
- allows for payments to be made directly to a user's xed address (without user interaction).

How does it do it?

zk-SNARKS
Zero Knowledge Succinct Non Interactive
Arguments of Knowledge

Allows to:

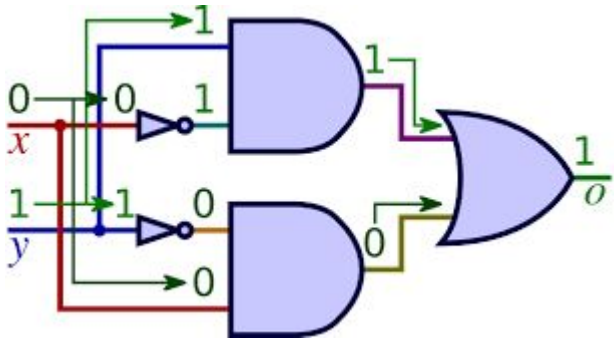
- hide transaction value inside the commitment
- split and merge transactions

Use of zk-SNARKS for Bitcoin also suggested by DFKP13

A few things about zk-SNARKS

Create efficient proofs for NP statements

- construct an arithmetic circuit for the statement to be proved

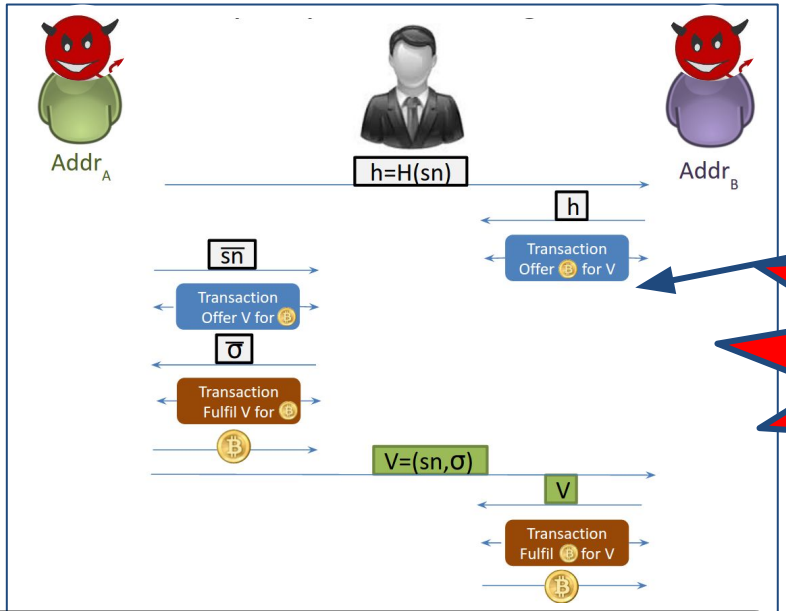


How are they different from NIZKs?

- Both need trusted setup & provide same guarantees (completeness, proof of knowledge, ZK)
- Proof length depends only on the security parameter and verification time on instance size (not on circuit)
- Security relies in very strong assumptions (knowledge-of-exponent)

thank you!

Resisting DoS and Sybil Attacks.



Intermediary has to front bitcoins for exchange

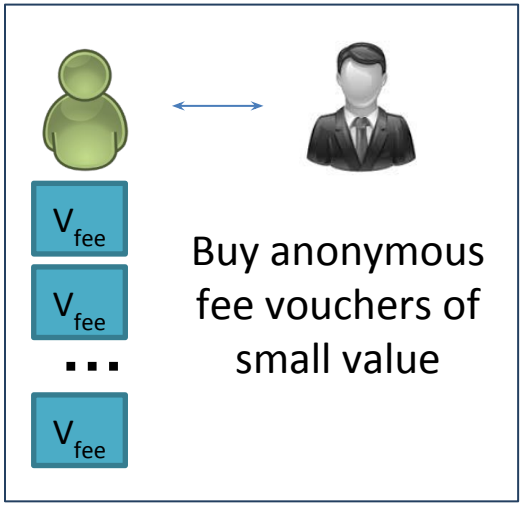
DoS risk!

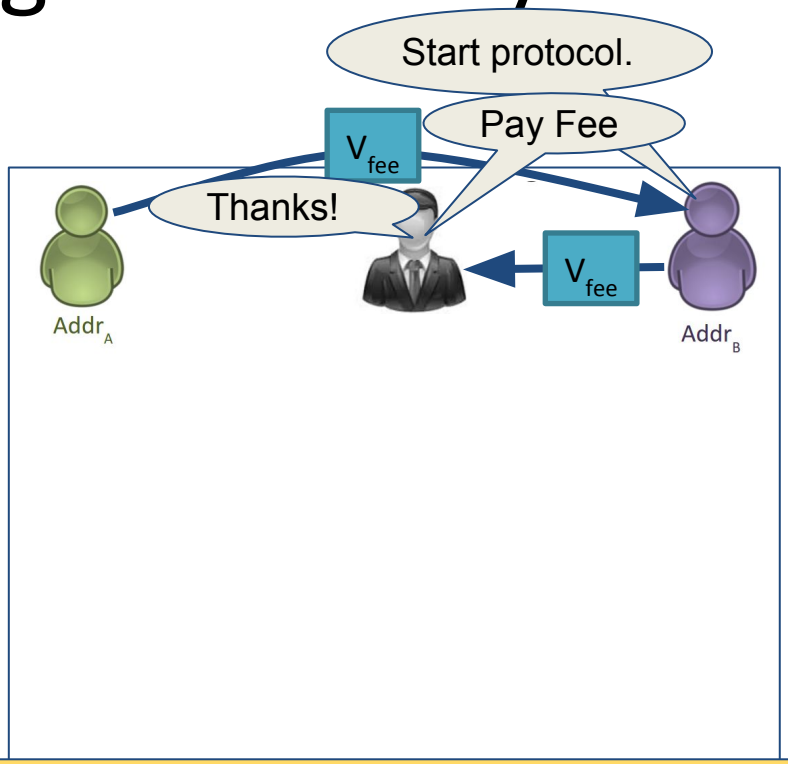
Solution! Make Bob pay a fee to start the protocol*,
Bob can pass this fee onto Alice,
...but how to do this anonymously?

Anonymous fee vouchers.

* Inspired by the fees used by XIM [1] to resist DoS and Sybil attacks.
[1]: 'Sybil-resistant mixing for bitcoin.' Bissias, Ozisik, Levine, Liberatore.

Resisting DoS and Sybil Attacks.





Also protects against Sybil attacks since sybils must now pay a fee.

* Inspired by the fees used by XIM [1] to resist DoS and Sybil attacks.
[1]: 'Sybil-resistant mixing for bitcoin.' Bissias, Ozisik, Levine, Liberatore.

Zerocoin - main idea

Implementing BB with Bitcoin

Recall how Bitcoin transactions work

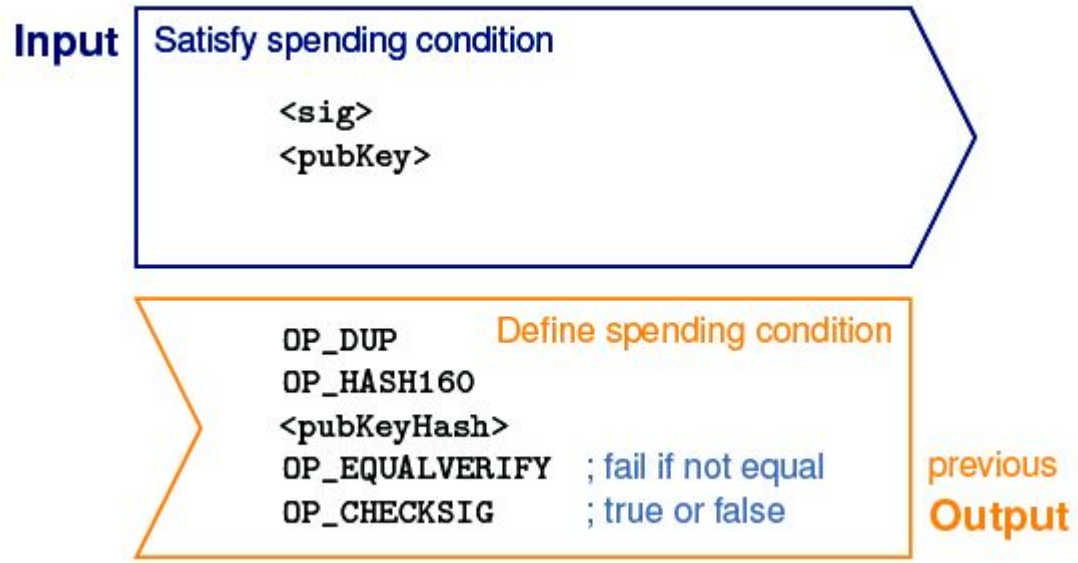


Image by Rainer Bohme

Zerocoin - main idea

Implementing BB with Bitcoin

Minting a zerocoin of value d: Alice creates a transaction and includes commitment C to output. The bitcoin value is put into escrow

Spending a zerocoin: Alice creates a transaction that spends any unclaim bitcoin on escrow to Bob and also includes (SN, π). Successful if π verifies.