The Fight for Privacy: Anonymization Techniques, Protocols, and Altcoins

Max Fang
(significant contribution by Philip Hayes)

ABOUT MAX

BERKELEY SPRING 2019

Overview

- Advisor **World Economic Forum**: Blockchain & Distributed Ledger Technology
- Cofounder **Dekrypt Education**
  - Teaching executives since March 2017
- Adjunct Professor **Berkeley Law:** Blockchain For Lawyers
- Writer, in progress **book** on blockchain use cases
- Advising and consulting
- Independent Research (Blockchain & Related)

Previously

- Cofounder, President **Blockchain at Berkeley** ('15-'18)
- Cofounder **Blockchain Unlocked**
- Cofounder **Blockchain Fundamentals** Fall 2016
  - Now on EDX: 80,000+ enrolled

BerkeleyLaw
UNIVERSITY OF CALIFORNIA

BLOCKCHAIN
AT BERKELEY

ABOUT MAX

BERKELEY SPRING 2019

Also

- Computer Science & Economics **UC Berkeley**
- Developer Advocacy **Lightning Network**
- Software Engineering **ChangeTip** 2015-16
- GPU Cryptocurrency Mining Feb 2014
  - **5 years** Bitcoin / blockchain experience

BLOCKCHAIN AT BERKELEY

BITCOIN Association of Berkeley

LIGHTNING

Past clients & speaking:

Bundesministerium für Wirtschaft und Technologie

Qualcomm

ExxonMobil

inspur 浪潮

Google

IDEO

Mercedes-Benz

WANXIANG BLOCKCHAIN 万向区块链

AB InBev

TOYOTA MATERIAL HANDLING

Stanford Blockchain Conference

Tsinghua University

BITCOIN HK

MALLPLAZA

中国科学院大学 University of Chinese Academy of Sciences

LAZARD ASSET MANAGEMENT

中国邮政储蓄银行 POSTAL SAVINGS BANK OF CHINA

BLOCKCHAIN AT BERKELEY

Lecture Outline

Goal: Get up to date with the State of the Art in Bitcoin mixing

Anonymity Basics

Deanonymization techniques

Anonymity through Mixing

Decentralized Mixing

Anonymous Altcoins

Conclusion

Anonymity Basics

Anonymity Basics

Blockchains are not anonymous by default.

- Intuition: Blockchains take a central database and distribute it
  - However, this means that you now have no access control
- All of the data is public by default
  - Private blockchains are slightly more anonymous since only a few members have access to the database

Most blockchains are **pseudonymous** - we use an identity that is not our real identity (e.g. your Bitcoin address)

- Our **pseudonyms** may or may not be "linked" to our real identity

Anonymity Basics

"Linking" in the context of anonymity is associating a real world identity to a pseudonym. This is also called **deanonymization**

- In Bitcoin: an identity and an **address**
- In Ethereum: an identity and an **account**

Bitcoin best practice achieves a small degree of anonymity

- Best practice: Never reuse your pseudonyms!
  - Generate a new address every time you receive Bitcoin
  - Like creating a new reddit account for every single comment
  - But basic analysis renders this technique ineffective
- Not possible in Ethereum, since it is account-based (not UTXO based)

Anonymity Basics

Anonymity isn't absolute (not a clear yes or no)

- The **"degree of anonymity"** (or sometimes "**level of anonymity**") is defined by how difficult it is to associate your pseudonym with your real world identity.

A high degree of anonymity allows you to reasonably expect having achieved **privacy.** But why is this important?

"Anonymity is only for buying drugs, right?"

Imagine these scenarios in a blockchain-based financial world.

'Bob's Burgers'

You make a purchase at Walgreens. Your cashier looks you up on blockchain.info and sees 20 purchases a month to the address publically labeled "Bob's Burgers," but everyone knows that that's the hidden name for the internet's biggest porn site.

Extreme example - blackmail: The same store employee also sees that you're sitting on a stash of $60 million in Bitcoin. When they kidnap your mother next week, they know exactly how much money to blackmail you for.

"Anonymity is only for buying drugs, right?"

Example: Getting paid back by a friend

A restaurant refuses to split the bill, and you volunteer to foot it. Your friend send you some Bitcoin. Later, you go to Bob's Burgers to make a purchase with your friends' Bitcoin, but they don't accept your payment because "your money is associated with drug dealers."

Fungibility is the idea that every unit of a currency must be equal in value to every other unit

- Crucial property of currency

NOV 13, 2013 @ 08:17 AM 38,863 VIEWS The Little Black Book of Billionaire Secrets

Sanitizing Bitcoin: This Company Wants To Track 'Clean' Bitcoin Accounts

Kashmir Hill, FORBES STAFF

Welcome to The Not-So Private Parts where technology & privacy collide FULL BIO

Alex Waters, Matt Mellon, and Yifu Guo, of Coin Validation

Source: Forbes on "Coin Validation"
http://www.forbes.com/sites/kashmirhill/2013/11/13/sanitizing-bitcoin-coin-validation/#6bb370ed6a45

"Anonymity is only for buying drugs, right?"

Example: Businesses on the blockchain

You've just founded a hot new startup run purely on the blockchain - BitBlockBaseCoinPay.cash. You want to keep up to date with your competitor CoinBitBlock.pay so you purchase their product. Except now they know all of your operational expenses, how much revenue you have, who your customers are, and your secret business strategy.

Conclusion: A lack of anonymity means everyone you've ever transacted with gets to see how you've spent your money in the past and forever into the future.

Source: CoinTelegraph

Anonymity and Ethics

Anonymous cryptocurrencies can indeed be used for money laundering and online drug purchases.

- Partial solution: the interfaces between cryptocurrencies and fiat currencies are highly regulated
  - Recall AML/KYC from last lecture: can trade cryptocurrencies almost anonymously but can't touch USD/GBP/EUR without a picture of your passport
- Hard to implement "morality" at a technological level
  - Moral and immoral use cases look identical from a technological standpoint
- Do the positive benefits to society outweigh the costs?
  - Example: Tor
    - Created by the U.S. government. Makes it difficult for the officials to monitor web traffic, but they've found other ways
    - Enables free speech for reporters in oppressive regimes

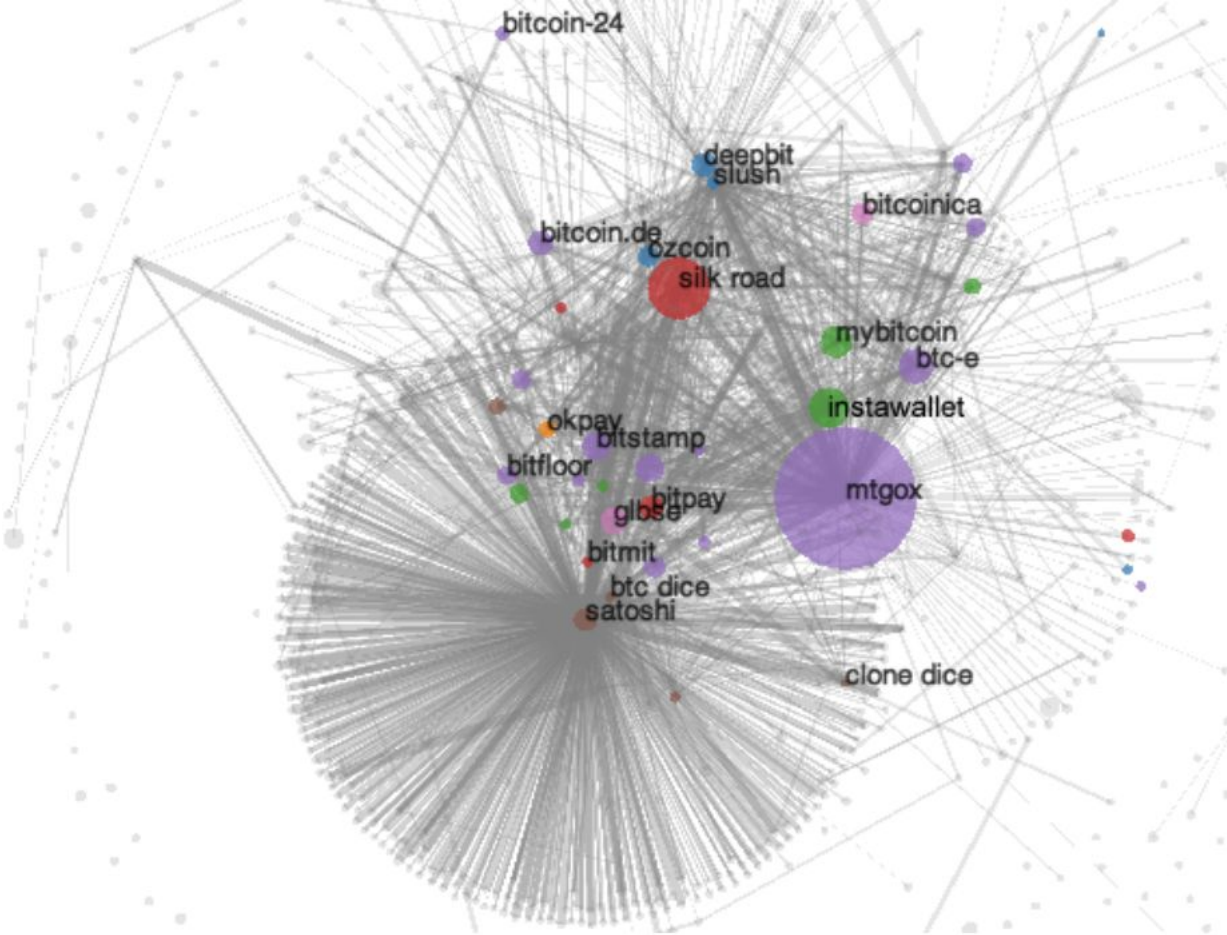
Basic Deanonymization Techniques

Deanonymization via Transaction Graph Analysis

Transaction Graph Analysis: Analyzing the graphs of transactions in the blockchain

Goal of deanonymization: **Link** an entity's real world identity with their pseudonym(s)

Clustering: Attributing a **cluster** of addresses to the same entity



Bitcoin's transaction graph in 2013.
A Fistful of Bitcoins: Characterizing Payments Among Men with No Names (Meiklejohn et al)

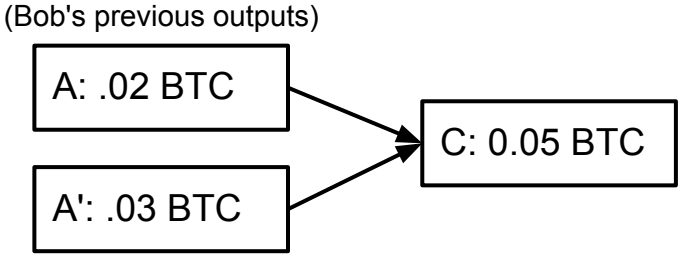
Clustering

Two main heuristics to associate two addresses:

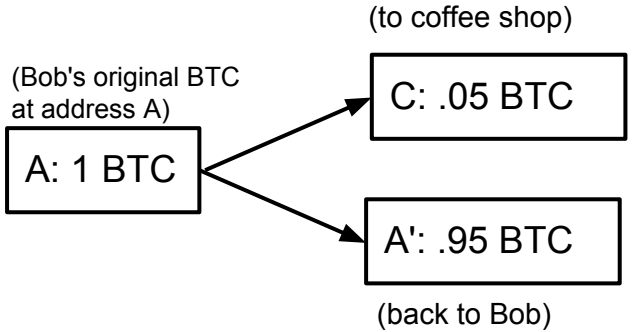
1. **Merging of transaction outputs**
    a. Occurs when there are multiple inputs to a transaction
    b. Fairly reasonable assumption that the two input addresses are paired by the same entity
        i. Rarely do people conduct joint payments
2. **Change addresses**
    a. Transaction is split into 0.95 and 0.05 amounts
        i. One of them must be a change address unless two items were purchased jointly
    b. Helpful heuristic: Change addresses are usually newly generated - never before seen on the blockchain

In both cases, if address **A** was known to be owned by Bob, we now know that address **A'** is also owned by Bob.

Case 1: Buying coffee of cost 0.05 BTC with 0.02 BTC and 0.03 BTC UTXOs. *A and A' merging into one output links them together.*



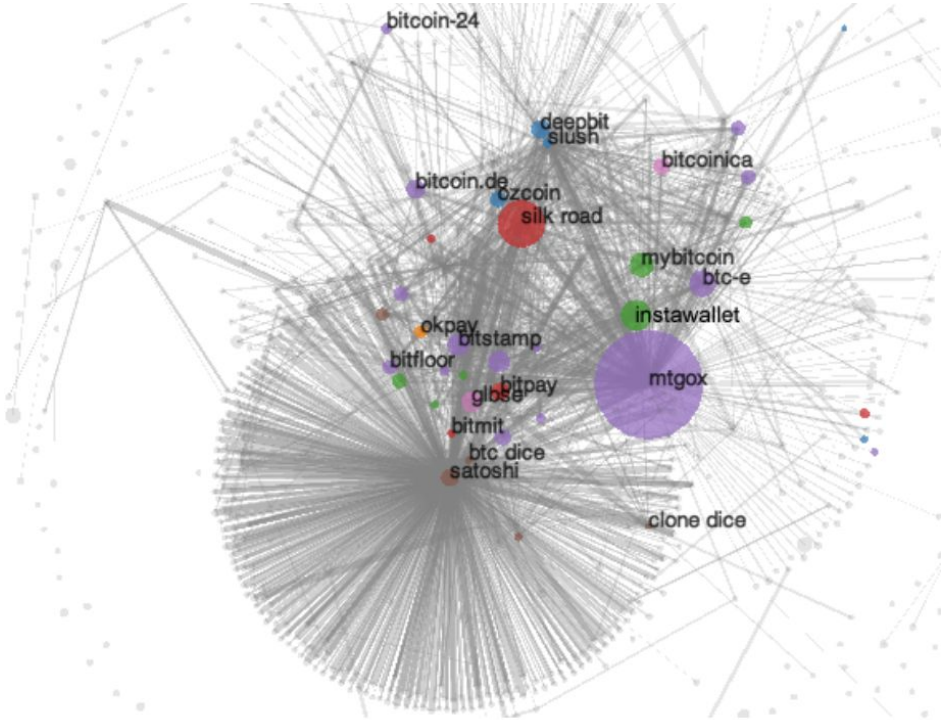
Case 2: Buying coffee of cost 0.05 BTC with a 1 BTC UTXO. *Identifying the change address links addresses A and A' together.*



Identifying services

Several techniques to identify clusters with the real world identities of businesses:

1. **Tagging by transacting**
   a. Go to online service (e.g. Coinbase) and make a transaction with them
   b. Wait for address to be merged with rest of the cluster
2. **Infer by looking at activity**
   a. In 2013, Mt. Gox was large part of ecosystem
      i. Large volume (large purple dot)
   b. SatoshiDice was a gambling site allowing smaller denominations
      i. Small volume (small dot)
      ii. Lot of transactions

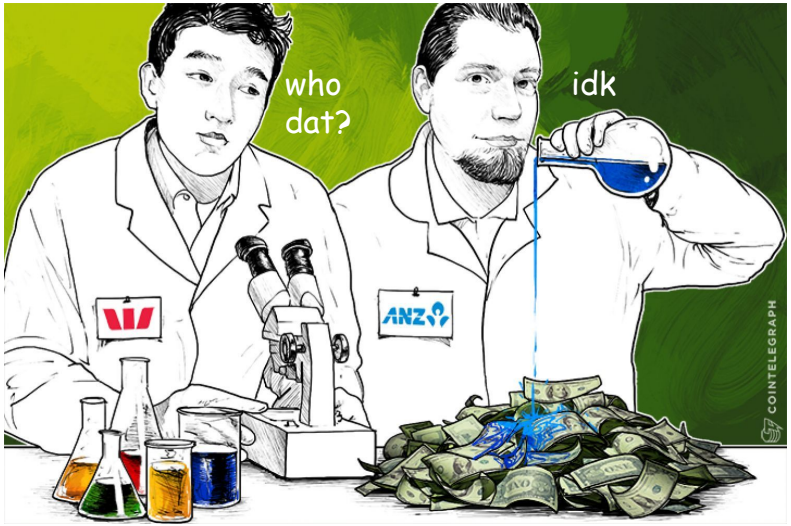


Bitcoin's transaction graph in 2013.
A Fistful of Bitcoins: Characterizing Payments Among Men with No Names (Meiklejohn et al)

Identifying individuals

Several techniques to associate addresses with individuals:

1. **Sending them Bitcoin**
   a. Obviously, they need to reveal an address
2. **Carelessness**
   a. Posting your Bitcoin address publicly anywhere (like on forums) reveals at least one address
3. **Service providers**
   a. Ex. Skry (previously Coinalytics)



Source: CoinTelegraph

Compliance/AML

Skry

Expose funds derived from illicit activities and detect complex money laundering activities.

Source: skry.tech ("Bloomberg for Bitcoin")

Taint analysis

Each circle is an address.

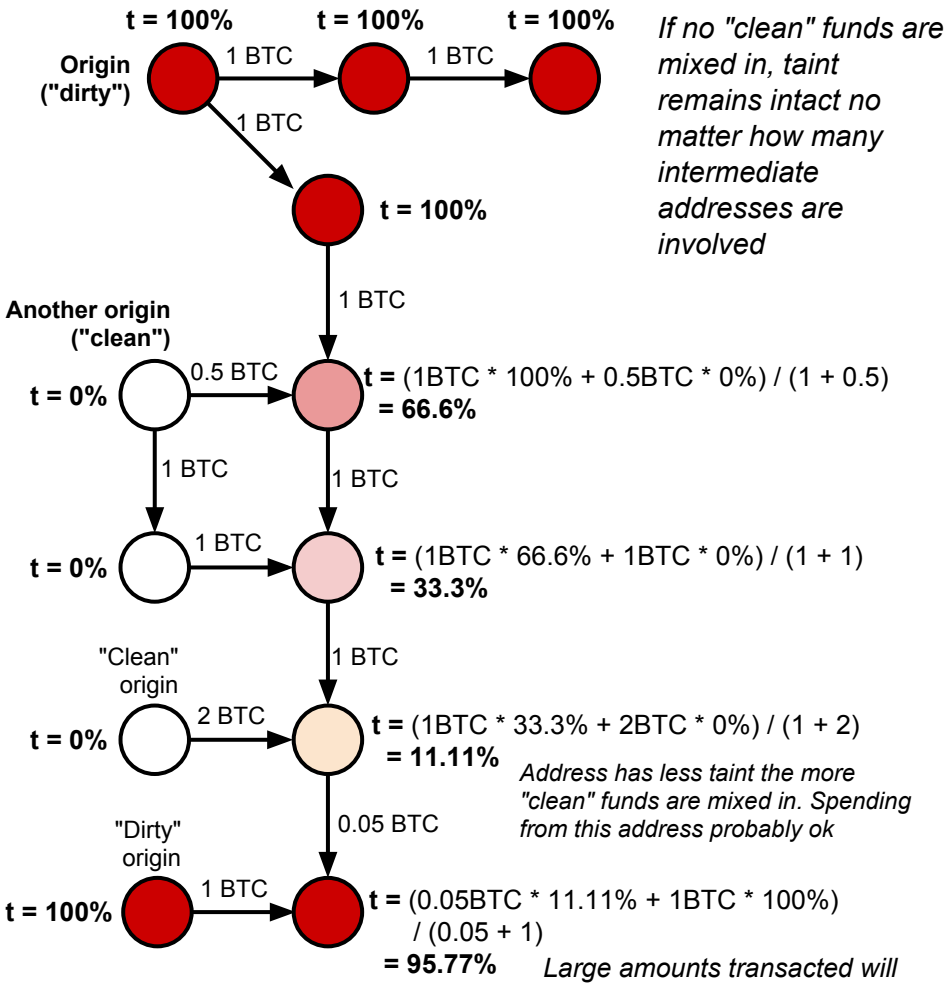
Let **t** denote the "taint" at that address.

Taint is the percentage of funds received by an address that can be traced back to another address

Taint analysis can reveal useful information

- See whether money came from a 'tainted' source
- Example: tag a known "bad" address
  - E.g. Silk Road
  - Taint analysis ruined Ross Ulbricht's defense that his huge Bitcoin stash was obtained legitimately!

Naive anonymization strategy: send all your coins to a bunch of fresh addresses (**manual mixing**). Taint analysis is why manual mixing doesn't work!



Taint Analysis 1dice6GV5Rz2iaifPvX7RMjfhaNPC8SXH

Taint is the % of funds received by an address that can be traced back to another address.

This pages shows the addresses which have sent bitcoins to 1dice6GV5Rz2iaifPvX7RMjfhaNPC8SXH. The data can be used to evaluate the anonymity provided by a mixing service. For example Send Coins from Address A to a Mixing service then withdraw to address B. If you can find Address A on the taint list of Address B then the mixing service has not sufficiently severed the link between your addresses. The more "taint" the stronger the link that remains.

Received (Origin) Taint

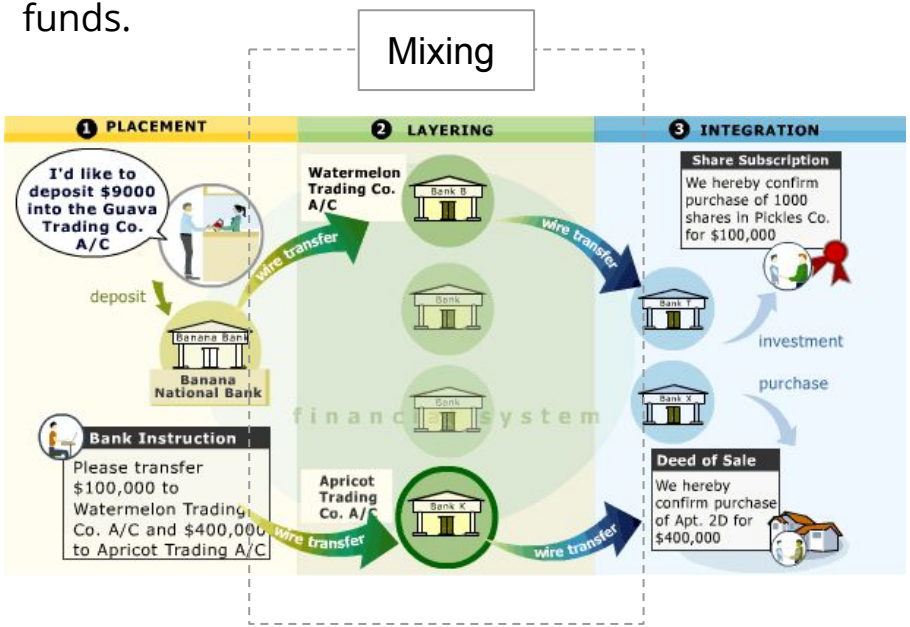
Branch	Address	Taint (%)	Count	Top IPs
21	17V7mV5yWgzkWVB6VGzJh6jiVcAYJ1xU8t	5.709493158%	48	
4	12p1dnSn11aXS1hBjt9cscZNTGSJ56YDQM	5.4376125314%	56	
3	1Lpn1Bhp8jieEGyraJ5koPrv7dEatgkB5k	5.3696423747%	10	
2	1P3TjAGvaqdTT2so8xm5MxXu55SCVss59Y	2.7188062657%	6	
2	1HG2RQWwiqr479GKhbykWn6FdbdQoBpU6H	2.7188062657%	66	
2	12U8dsx3grbyBDRjR7AQpvD2eedgqvWnyo	2.7188062657%	6	
3	1bankkjx5E9Xqd5... (Satoshi Dice Change Address)	2.497099566%	9	
5	1dice97ECuByXAv... (SatoshiDICE 50%)	2.2296799195%	24	

Taint analysis tool on Blockchain.info

Anonymity through Mixing

Mixing

Mixing: Making transactions with the intention of concealing the origins of your funds.



source: http://casiinoo.blogspot.com/

Traditional Mixing / Money Laundering:

Create hundreds of fake “shell” companies, which don’t do anything or own any assets, but *look* like they do (according to the accounting books and tax returns).

Over time, deposit “dirty” funds into shell corps. (Placement).

Shell corps. write off deposits as purchases, investment, etc… to make deposits look real.

Shell corps. further **obfuscate by sending funds to *other* shell corps** (Layering).

Finally, criminal org. spends “clean” money on luxury goods, e.g., diamonds, cars, real estate (Integration).

Mixing on blockchains harness the same idea.

A Formal Framework for Anonymity

Def. An **anonymity set** is the set of pseudonyms between which an entity cannot be distinguished from her counterparts

Main goal of mixing: We want our anonymity set to be as large as possible

- Conducting multiple rounds of mixing exponentially increases our anonymity set
- If one round of mixing makes you indistinguishable among **N** peers, then size of anonymity set is **N** for one round, $\mathbf{N^2}$ after two rounds, $\mathbf{N^3}$ after three, etc.
- However, the size of the anonymity set is bounded by real world constraints
    - e.g. the above assumes peers themselves continue to mix

The larger the anonymity set, the harder it is to deanonymize, or "re-link", pseudonyms to identities.

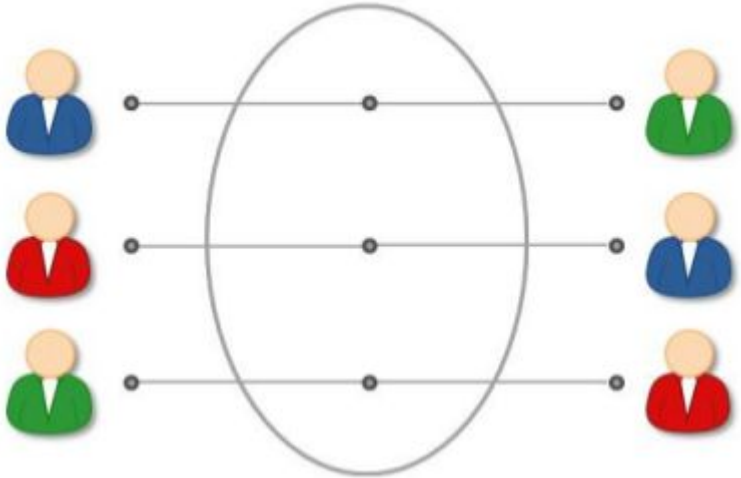
- Ideally, it is hard for **anyone** to link identities to addresses

Additional desirable properties

- **Trustless** (No counterparty risk)
    - Want to ensure that our funds can't stolen while mixing
- **Plausibly deniable**
    - It shouldn’t be obvious from transaction history and any other data traces that you’re mixing; i.e. your activity should look just like normal activity

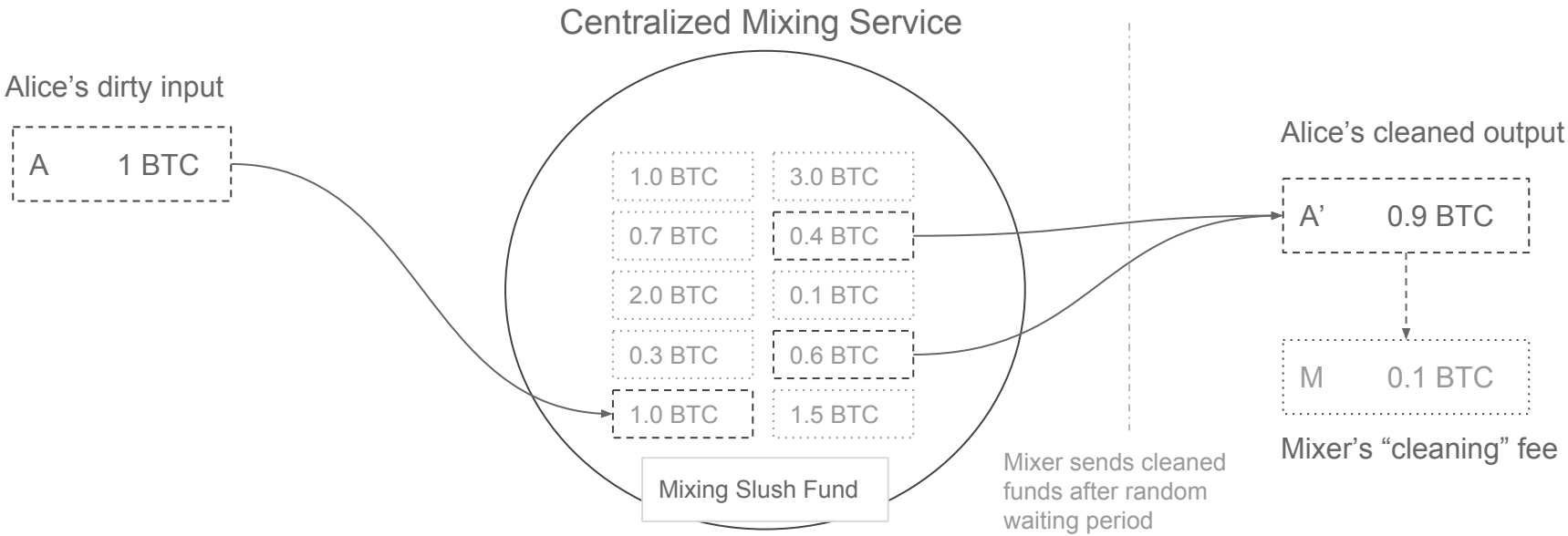
Types of Mixing

- Centralized Mixers
- Altcoin Exchange Mixing
- Decentralized Mixing Protocols
- Privacy-focused Altcoins



Centralized Mixers

Send coins to a Trusted Third Party (TTP), TTP sends (hopefully) unlinked coins to you sometime in the near future.



Centralized Mixers - Issues

Counterparty Risk: Mixer could steal funds; have to *trust* that it won't.

Logging Risk: Mixer could be logging who it received dirty funds from and where it sent the cleaned funds to.

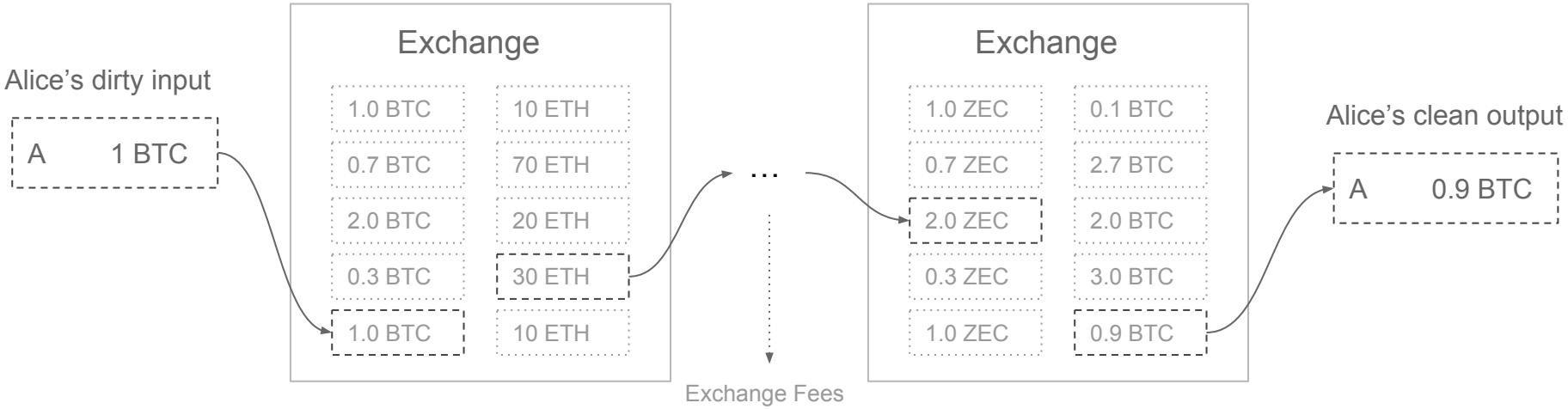
Centralization Risk: Single point of failure. Single target for hacking. Adversary (e.g. Government) installs its own logging or sends a takedown notice and seizes control of mixer.

Examples of centralized mixing protocols

- **Mixcoin**
  - Relies on the TTP accountable
- **Blindcoin**
  - Same as Mixcoin, but TTP is blind - cannot infer the permutation of mixes

Altcoin Exchange Mixing

Idea: Send dirty funds through several layers of altcoin ⇐⇒ altcoin exchanges to obfuscate money trail.



Altcoin Exchange Mixing - Issues

Pros:

+ Adversary would have to trace transaction chain through **several disparate blockchains** and exchanges.
+ **Better plausible deniability** -- looks like normal currency exchanging.

Cons:

- Rely on exchanges keeping transaction mappings hidden
- **Counterparty risk**: Exchange gets hacked ⇒ Lose money in transit
- (U.S.) Exchanges usually require personally identifiable information and follow **KYC/AML**.

Decentralized Mixing Protocols

Decentralized Mixing Protocols

Idea: Remove counterparty risk and avoid fees by taking out the middleman (centralized mixer).

Proposition: Create a network of peers outside of Bitcoin network who cooperate to make transactions which mix their coins, without relying on a trusted third party.

Can this be done?

Dmix Research Project

Well, this is the question we (Max Fang and Philip Hayes) sought to answer.

Dmix Project: Build a **trustless**, **decentralized Bitcoin** mixer that maintains **plausible deniability**.

Additional goals:

- **Low fees**
  - Mixing shouldn't be cost-prohibitive; would be impractical
- **Bitcoin-compatible**
  - Must be able to be implemented in Bitcoin, e.g. doesn't use pairings
  - Not interested in purely theoretical exercises

So, let's build Dmix!

Decentralized Mixing Protocols - Nuances

Additional considerations for designing a good decentralized mixing protocol

A mix is comprised of inputs and outputs:

- One input and one output are owned by the same entity, and the goal of the mix is to hide the **mapping** from all inputs to all outputs.

Def. Correctness: Coins must not be lost, stolen, or double-spent. The mixing is truly random and must eventually succeed in mixing or returning the funds of honest users (resilient against DoS attacks).

Adversarial models:

- **Passive adversary**
  - Not a part of the mix
  - Basic anonymity prevents passive adversaries from learning the mapping
- **Semi-honest adversary**
  - Part of the mix
  - Correctly follows the protocol but attempts to deanonymize the mix by analyzing the procedures of the mix.
- **Malicious adversary**
  - Part of the mix
  - Not bound by the protocol specifications; may actively deviate from the protocol and attempt to steal funds
  - May send false messages, abstain communications, etc.

Decentralized Mixing Protocols - Nuances

Sybil resistance in the context of decentralized mixing has a two part definition:

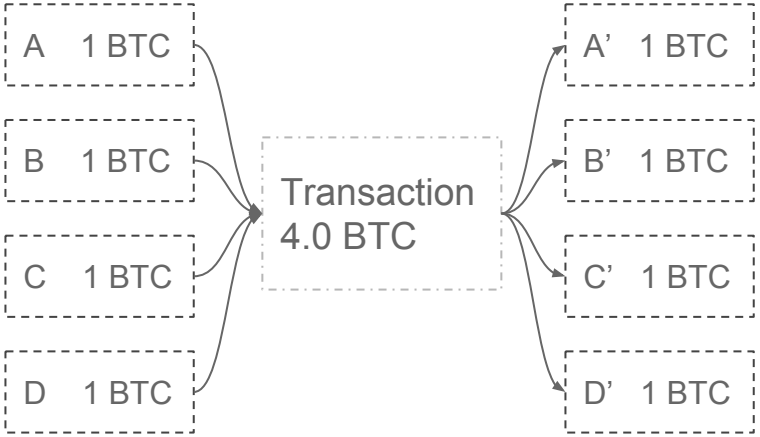
1. Resistance to stealing funds
    - Can't rely on 'partial' threshold cryptography to enforce correctness (e.g. m-of-n multisig such that m < n).
    - Protocol must execute correctly (no funds are stolen) even if all other peers are malicious adversaries
2. Resistance to deanonymization
    - **Weak**: Participants outside the mix cannot determine the mapping of inputs to outputs, but participants within the mix can.
        - Only requires one semi-honest adversary to break anonymity
    - **Strong**: Even participants within the mix do not know the mapping of inputs to outputs
        - ***However, a high proportion of Sybil peers reduces the anonymity set.***

Mixing Caveats

- **Side channel attacks**
  - Generally, we want to use TOR for everything
  - However, TOR exit nodes may be adversary-controlled
- **Analyzing transaction amounts**
  - Easy to identify input and outputs (E.g. 1337.6969 BTC in -> 1337.420 out: hmmmmm □)
  - Solution: Always use uniform transaction amounts (like 1 BTC, 0.1 BTC)
  - All transactions going through all mixes would look the same
  - For this reason, fees should be all or nothing
- **Timing correlations**
  - e.g. Linking the input to a mix with the immediately subsequent output of the mix
  - Solution: adding (random) delays, concurrent ins and outs
- **Network-level deanonymization** (transaction propagation)
  - "The first node to inform you of a transaction is probably the source of it."
  - Solution: Network anonymization techniques like Dandelion

Protocol - CoinJoin (2013)

Alternative Approach: Mix together coins in a single n-of-n multisignature transaction.



Pros:

\+ **Trustless:** Funds can't be stolen

Cons:

- **Anonymity not secure against passive adversary (mix facilitator)**
  - Best existing implementation for executing the protocol is via a **centralized server**; assumes private and anonymous communication channels for submitting output addresses. E.g. vulnerable to **traffic analysis**
- **Not plausibly deniable;** very easy to spot on the blockchain since n-of-n multisignature transaction where n is usually large. (Can be fixed with Schnorr sigs)
- **Not DoS attack resistant;** only needs 1 malicious node to start protocol and then halt halfway through to disrupt.

CoinJoin: https://bitcointalk.org/index.php?topic=279249.0

CoinJoin ⇒ DASH

DASH (formerly DarkCoin) is a privacy-centric cryptocurrency that employs a network of Masternodes to perform privileged actions such as voting on proposals, instantly confirm transactions, and **mix the coins (by default) of all network participants.**

Pros:

- Uses CoinJoin for mixing: trustless
- No issue of plausible deniability with using CoinJoin since almost everyone on the entire network is participating in CoinJoin transactions

Cons:

- Masternode network itself must be secured - can pay 1000 DASH per masternode to hypothetically acquire a large number of them

CoinShuffle (2014)

Builds on **CoinJoin** to build a better communication protocol, is **decentralized**

Uses a **decryption mixnet** to jointly compute the input/output shuffling

Pros:

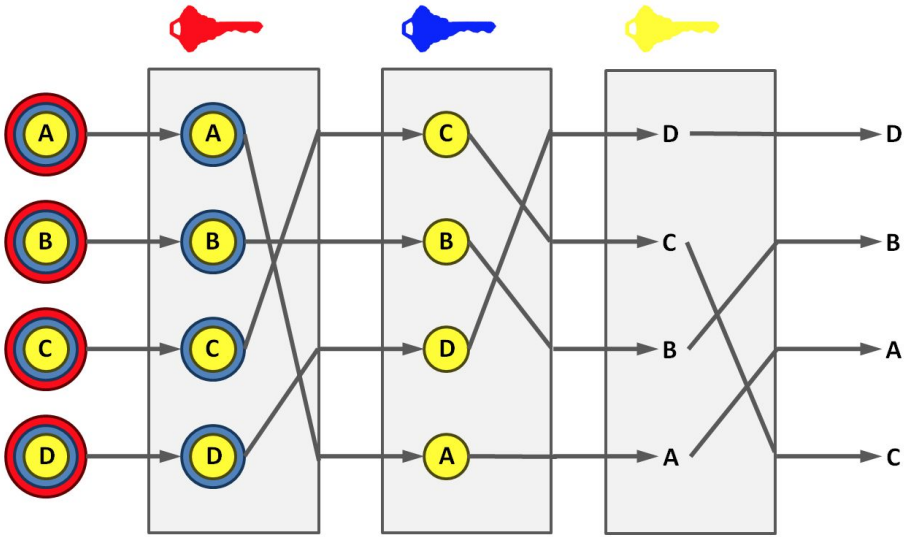
- Uses an "Accountable Anonymous Group Messaging" protocol called **Dissent** to resolve issue of **traffic analysis attack**
- Achieves **anonymity against mix facilitator**, because it is decentralized
- Achieves strong **Sybil-resistance against deanonymization**

Cons:

- Suffers from all the drawbacks of CoinJoin
  - Vulnerable to **deanonymization via Sybil attack**
  - Vulnerable against **DoS** attack
- The last peer in the decryption mixnet is in a unique position to determine the outcome of the shuffling

CoinShuffle - Decryption Mixnet

- Output addresses encrypted layer by layer among the public keys of the other peers
- Each peer decrypts, applies a private permutation to the decrypted blob, and passes it to the next peer.
- Process repeats until the last peer decrypts the last layer and thus obtains the final permutation.



Protocol - CoinParty (2015,2016)

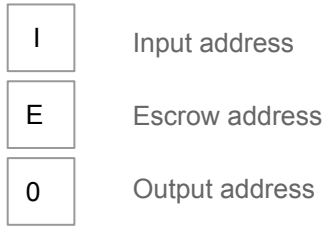
Idea: Decentralized mixing protocol but with **better deniability**. Want transactions to look the same as normal Bitcoin transactions to passive observers.

Is this possible?

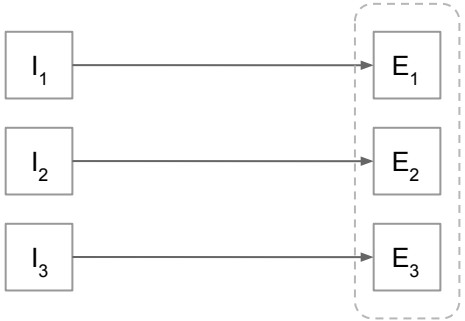
CoinParty lets us do this, but sacrifices some protocol security.

CoinParty: http://www.martinhenze.de/wp-content/papercite-data/pdf/zmh+16.pdf

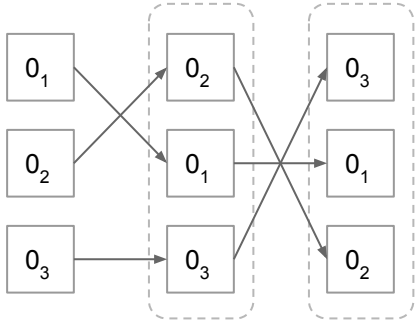
Protocol - CoinParty (2015,2016)



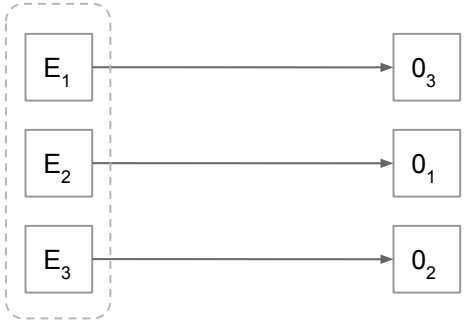
Peers generate escrow addresses. Escrow addresses require ⅔ consensus to spend.



Peers perform secure multi-party shuffle on output address ordering.



If protocol executed correctly, peers agree to transfer funds out of escrow addresses to designated outputs.



1 COMMITMENT

2 SHUFFLE

3 TRANSACTION

Protocol - CoinParty (2015, 2016)

The main stages of CoinParty, in more detail:

1. **Commitment:** Sending funds to escrow addresses
    a. Each mixing peer uses **P**seudo**r**andom **S**ecret **S**haring (PRSS) to obtain a share of the private key. PRSS achieves a low communication overhead
    b. Using the private key, mixing peer computes their share of the public key
    c. Broadcast their share of the public key to the other mixing peers and jointly reconstruct the escrow address
    d. Complete this process once for every input peer.
2. **Address Shuffling:** Like CoinShuffle, jointly compute input/output shuffling, fixes shuffling by verifying correct decryption via checksums
3. **Transaction:** Use threshold signatures to sign transactions
4. **Error and Reversion:** Punish those who tried to cheat the protocol

Protocol - CoinParty (2015,2016)

Pros:

+ **High plausible deniability;** transactions on the blockchain look just like “normal” Bitcoin transactions.
+ **Decent efficiency;** requires 2 transactions on the blockchain per input peers.
+ **Bigger anonymity set**; large number of "normal" Bitcoin transactions with the same amount is orders of magnitude more anonymous

Cons:

- **Reduced protocol security;** escrow funds controlled by ⅔ threshold signature scheme.
- **Vulnerable to Sybil Attack;** malicious peer can spawn several fake peers, join mix group, overthrow ⅔ threshold, and steal mix group’s funds.

JoinMarket (2015)

All of the previous protocols have a **liquidity problem** - in the real world you'll usually only mix with other people who have dirty coins

Idea: Create market of liquidity providers who are willing to mix their coins for a fee.

Since market makers take almost no risk, mixing fees are typically very small.

Issues:

- Anonymity set still fairly small
- Deanonymizing entire system would require only $32,000 (recoverable after attack) with success rate of ~90% (Möser, Böhme)

JoinMarket Orderbook

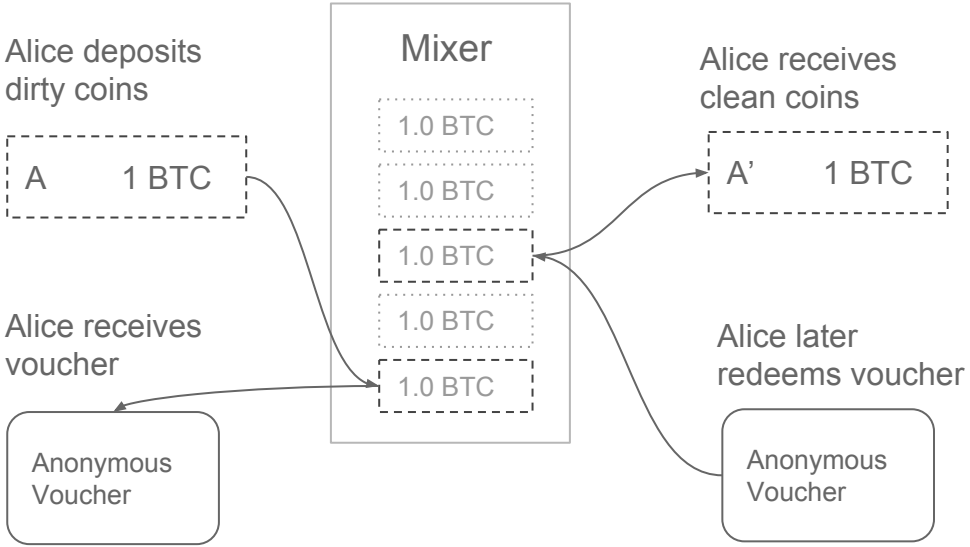
142 orders found by 66 counterparties

Type	Counterparty	Order ID	Fee	Miner Fee Contribution / BTC	Minimum Size / BTC	Maximum Size / BTC
Absolute Fee	J5CZTub55wvWFZBu	0	0.00000969	0.00000000	0.00003830	0.00160000
Absolute Fee	J5CZTub55wvWFZBu	4	0.00001000	0.00000000	0.00003830	7.49206132
Absolute Fee	J5CZTub55wvWFZBu	25	0.00001000	0.00000000	0.00003830	0.01200000
Absolute Fee	J54ipjp2Diz9XqMS	1	0.00001750	0.00000000	0.00010000	0.99999999
Absolute Fee	J5CZTub55wvWFZBu	2	0.00002700	0.00000500	0.00003830	7.08951594
Absolute Fee	J5CZTub55wvWFZBu	18	0.00002818	0.00000000	0.00003830	0.00971051
Absolute Fee	J54ipjp2Diz9XqMS	2	0.00002928	0.00000000	1.00000000	1.99999999
Absolute Fee	J523sac3EtDzLN8P	1	0.00002985	0.00000000	0.00002730	0.00976520
Absolute Fee	J5CZTub55wvWFZBu	14	0.00003000	0.00000000	0.00003830	4.14202467
Absolute Fee	J57wggYo1Q3uDjyV	0	0.00003100	0.00000100	0.00100000	1.44742679
Absolute Fee	J5CZTub55wvWFZBu	1	0.00003630	0.00000000	0.00003830	2.99999999
Absolute Fee	J54MdBzKZpz1xp4c	3	0.00003630	0.00000000	2.00000000	2.99999999
Absolute Fee	J5CZTub55wvWFZBu	17	0.00004100	0.00000100	0.00003830	5.09930543
Absolute Fee	J54exwiYnGkhJB9j	0	0.00004100	0.00000100	0.00100000	3.81806101
Absolute Fee	J5CZTub55wvWFZBu	5	0.00004287	0.00000000	0.00003830	1.99999999

JoinMarket: https://github.com/JoinMarket-Org/joinmarket
Möser, Böhme: http://weis2016.econinfosec.org/wp-content/uploads/sites/2/2016/05/WEIS_2016_paper_58.pdf

Fair Exchange Mixers

- **Fair-exchange** means it is trustless
- "Fair-exchange mixer" usually refers to the model of A paying B through an untrusted intermediary T



Recipient does not have to be depositor. I.e. A could be B

Enables Alice to deposit her dirty coins and receive clean, unlinked coins without revealing her identity

Relies on the assumption that if **enough transactions pass through the mixer at the same time, the mixer cannot tell which inputs map to which outputs**

TumbleBit: https://eprint.iacr.org/2016/575.pdf

Protocol - XIM (2014)

Similar to CoinSwap, uses an untrusted intermediary to create a **fair-exchange mixer**

- Builds on even earlier work on fair exchange (Barber, Shi et al. 2012)
- **Uses fees** to prevent **DoS** and **Sybil attacks**
- Creates a secure group-forming protocol for finding parties to participate in a mix

Con:

- Requires several hours to run because of the group-forming protocol

Protocol - BSC (2016)

Blindly Signed Contracts

Builds upon XIM to prevent the time-consuming group forming process

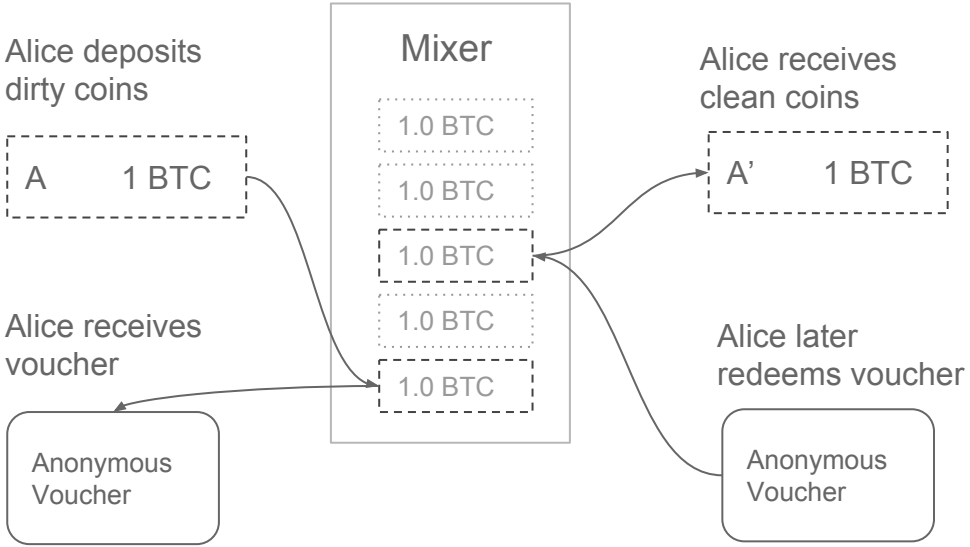
- Uses **anonymous fee vouchers** to resist DoS and Sybil attacks;

Con:

- Requires scripting functionality that is not supported by Bitcoin
  - Specifically, support of blind signatures constructable from ECDSA

Protocol - TumbleBit (2016)

Idea: Improve on **BSC** so the mixer **is Bitcoin-compatible**



State of the art in fair-exchange mixers

- Implements an "**RSA evaluation as a service**" protocol to make Blindly Signed Contracts **Bitcoin-compatible**
- Fairly feasible for real-world use, provided there is enough throughput

TumbleBit: https://eprint.iacr.org/2016/575.pdf

Protocol - CoinSwap (2013)

Using hashlocked, 2-of-2 multisignature transactions, you can securely swap your coins with someone else

Pros:

+ **Trustless**; mixer can't steal funds.
+ **Better plausible deniability** than the other fair-exchange mixers; passive adversary only sees 2of2 multi-signature transactions instead of complicated scripts

Cons:

- **Anonymity not secure against passive intermediary**
  - But the intermediary could be the person you are swapping with
- **Expensive;** uses 4 transactions per mix round

CoinSwap: https://bitcointalk.org/index.php?topic=321228.0

Dmix "Swinger Protocol" & Project Conclusion

The last iteration of Dmix project: **Swinger Protocol**

- Form pairs with your mixing group, designate one as the "husband" and the other as the "wife"
- Execute a decryption mixnet pairwise to obliviously obtain a designated pair that your pair shall swap with.
- Your "wife" is sent over to the designated husband. They perform CoinSwap to trustless exchange coins
- You were the designated pair for another pair; you receive an incoming wife from that pair. Your husband performs CoinSwap with the incoming wife.
- Abort protocol if no wife or more than one wife were received.

Nothing that currently exists meets the design goals set out for the Dmix project

- Swinger Protocol comes close, but has a lesser degree of anonymity than the naive mixing strategy of simply executing CoinSwap with random nodes on the Dmix network
- Forming mixing groups actually reduces the anonymity set since Sybils

Conclusion: Building a good decentralized Bitcoin mixer is **hard**.

Privacy-focused Altcoins

CryptoNote ⇒ Monero

Idea: Hide input/output mappings with Ring Signatures. Prove that your funds came from one of a set of outputs, without revealing which output.

- Choose some set of previous outputs to “mix” with. These are then bound with your outputs in a cryptographic ring signature.
- **Ring Signature:** In this context, prove you own one of the outputs without revealing which specific output.
  - Anonymity set is the set of outputs you're signing from
- Better plausible deniability since mixing enabled by default

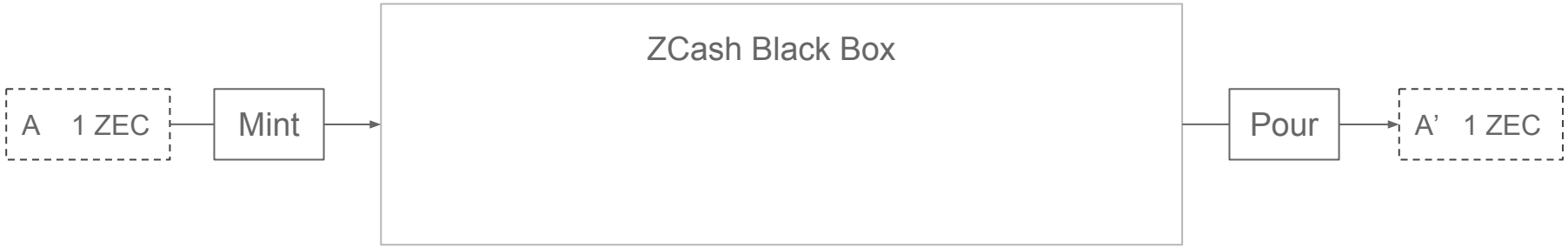
Issue: Monero doesn’t hide **transaction values** (yet). Adversary could potentially trace transactions by following likely value flows. **Temporal correlations** also pose an issue.

Issue: Decent anonymity set, but can we do better?

zk-SNARKs ⇒ ZCASH

Idea: Altcoin where transactions reveal *nothing* about input/output addresses AND input/output values.

Using **zero-knowledge Succinct Non-interactive ARguments of Knowledge** (zk-SNARKs) a.k.a. “Crypto Magic” we can create a system which supports **fully anonymous payments**.



ZCash

Pros:

+ **Fully Anonymous;** Assuming security of underlying crypto, blackbox transactions are anonymous. Anonymity set of entire blackbox history.

Cons:

- **Requires Semi-Trusted One-time Setup;** adversary with toxic setup parameters can mint coins without spending base coins. Can be somewhat mitigated with a secure multiparty computation setup.
- ~~**Resource Intensive;**~~ zk-SNARK proof system currently in use requires about ~~4 GB~~ **32 MB** of RAM and ~~2 minutes~~ **7 seconds** of computation on modern CPU to generate proofs for pour transactions. Sapling upgrades

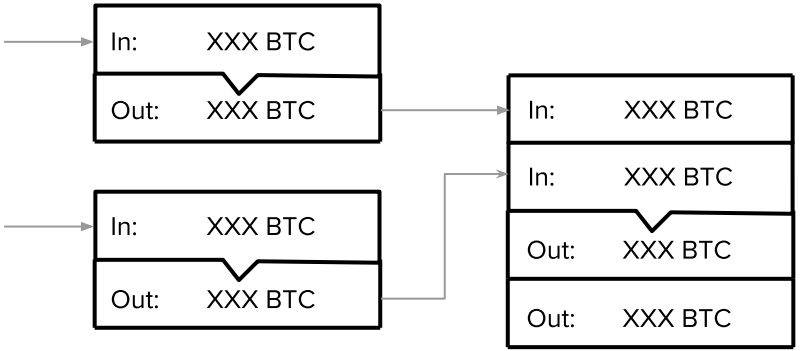
CT-based Techniques

CONFIDENTIAL VALUES / TRANSACTIONS

CT-BASED ANONYMIZATION

Idea:

- Modify transactions to **hide input and output values**
- Only transaction senders and receivers know the values.
- Every other person inspecting the blockchain can **verify correctness but not learn the values**.
- Does not hide sender and receiver pseudonyms!



Input and output **values are hidden** to other users.

AUTHOR: PHILIP HAYES

CONFIDENTIAL VALUES

CT-BASED ANONYMIZATION

Pedersen Commitments have a special property, they are *homomorphic* under addition

```
C(x1, r1) + C(x2, r2) = C(x1 + x2, r1 + r2)

C(x1, r1) - C(x1, r1) = 0
```

AUTHOR: PHILIP HAYES

PEDERSEN COMMITMENTS

CT-BASED ANONYMIZATION

How do Pedersen Commitments work? -- They use elliptic curve cryptography!

Assume some public generator points, G, H, prime orders, etc...

$$C(x, r) = x \cdot G + r \cdot H \quad \text{(create a commitment)}$$

AUTHOR: PHILIP HAYES

PEDERSEN COMMITMENTS

CT-BASED ANONYMIZATION

How do Pedersen Commitments work? -- They use elliptic curve cryptography!

Assume some public generator points, G, H, prime orders, etc...

$$C(x, r) = x \cdot G + r \cdot H$$

$$C(x1, r1) + C(x2, r2) = (x1 \cdot G + r1 \cdot H) + (x2 \cdot G + r2 \cdot H)$$

AUTHOR: PHILIP HAYES

PEDERSEN COMMITMENTS

CT-BASED ANONYMIZATION

How do Pedersen Commitments work? -- They use elliptic curve cryptography!

Assume some public generator points, G, H, prime orders, etc...

$$C(x, r) = x \cdot G + r \cdot H$$

$$\begin{aligned} C(x1, r1) + C(x2, r2) &= (x1 \cdot G + r1 \cdot H) + (x2 \cdot G + r2 \cdot H) \\ &= (x1 \cdot G + x2 \cdot G) + (r1 \cdot H + r2 \cdot H) \\ &= (x1 + x2) \cdot G + (r1 + r2) \cdot H \end{aligned}$$

AUTHOR: PHILIP HAYES

PEDERSEN COMMITMENTS

CT-BASED ANONYMIZATION

How do Pedersen Commitments work? -- They use elliptic curve cryptography!

Assume some public generator points, G, H, prime orders, etc...

$$
\begin{aligned}
C(x, r) &= x \cdot G + r \cdot H \\
C(x1, r1) + C(x2, r2) &= (x1 \cdot G + r1 \cdot H) + (x2 \cdot G + r2 \cdot H) \\
&= (x1 \cdot G + x2 \cdot G) + (r1 \cdot H + r2 \cdot H) \\
&= (x1 + x2) \cdot G + (r1 + r2) \cdot H \\
&= C(x1 + x2, r1 + r2)
\end{aligned}
$$

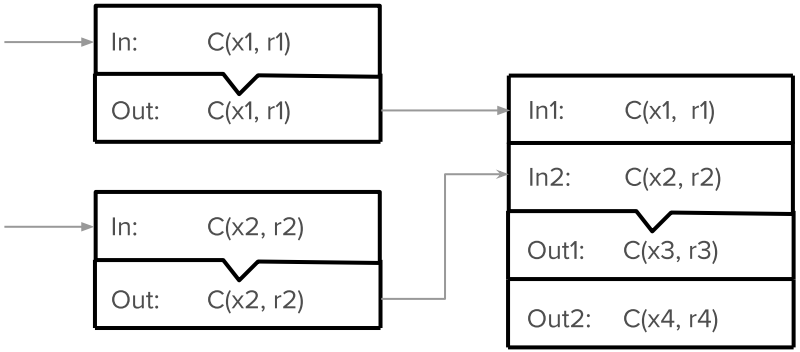
AUTHOR: PHILIP HAYES

CONFIDENTIAL VALUES

CT-BASED ANONYMIZATION

Idea: Use Pedersen Commitments for the transaction input and output values.

- **Transacting parties:** can see values and verify.
- **Other nodes:** just verify that **inputs - outputs = 0**,
  - never learn the transaction's actual input and output values.



Need to construct outputs such that:

```
x1+x2 = x3+x4, r1+r2 = r3+r4
```

Other nodes can verify:

```
(In1 + In2) - (Out1 + Out2) = 0
```

AUTHOR: PHILIP HAYES

CONFIDENTIAL VALUES

CT-BASED ANONYMIZATION

- In his paper, Greg Maxwell proposes adding a range proof to the construction and handles transaction fees.
- **Range proof:** allows other nodes to verify that a commitment value is in some non-negative range, e.g., $[0, 2^{64})$ without learning the actual value.
  - Attack: `(1+1) - (-5+7) == 0,` Discard the -5 output, keep 7 BTC!
  - Binary decomposition, Pedersen comms for each bit, ring signatures
- Builds a construction called a Borromean Ring Signature that provides this functionality with decent efficiency (32 bytes per commitment).

Too complex to describe in this presentation, but good further reading!

CONFIDENTIAL VALUES

CT-BASED ANONYMIZATION

Advantages:

1. Hides all transaction values ⇒ Improved anonymity and fungibility.
2. Improves effectiveness of coin mixing (CoinSwap, CoinJoin, etc...), since side channel attacks from transaction value leakage no longer give meaningful information.
3. Uses fairly vanilla cryptography--no bleeding edge zk-snarks or anything.

Disadvantages:

1. Does not directly hide transaction participants.
2. Requires interaction between sender and receiver; need to reveal commitments out-of-band...
3. Small-ish space penalty (~24B / txn)

MIMBLEWIMBLE

CT-BASED ANONYMIZATION

Idea: Strip down Bitcoin to bare minimum for payments system, combine Greg Maxwell's **Confidential Value Transactions** and **CoinJoin**, add **Block Merging**...

⇒ Mimblewimble!

1. Gives Bitcoin transactions near complete anonymity
2. Removes Bitcoin Script -- "too powerful"
3. Uses no exotic crypto other than discrete logarithms over elliptic curves, like Bitcoin.
4. Invented by alias *"Tom Elvis Jedusor"* a.k.a. French Voldemort

AUTHOR: PHILIP HAYES

MIMBLEWIMBLE

CT-BASED ANONYMIZATION

Consequences:

1. All transactions simply record input and output commitments (and range proof).
   a. No P2SH. No Bitcoin Script. Just simple payments.
2. We don't pay Bitcoins to an address.
   a. Perform a "ritual" (interactive protocol) between sender and receiver.
   b. Securely transfers value in input commitments to output commitments,
   c. Broadcast the input and output commitments to the network.

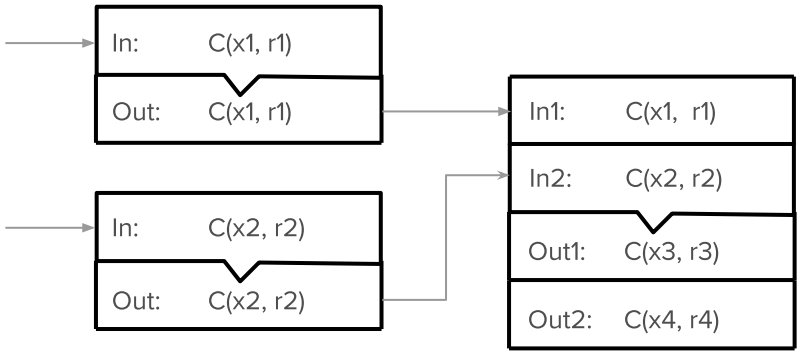
MIMBLEWIMBLE

CT-BASED ANONYMIZATION

Consequences:

All transactions simply record input and output commitments (and range proof).

- No P2SH. No Bitcoin Script. Just simple payments.



MIMBLEWIMBLE

CT-BASED ANONYMIZATION

Bitcoin:

Address model:

- use (private key, public key) pair that generates address as proof of ownership
- signing transaction with private key proves ownership

Mimblewimble:

Commitment model:

- c = C(x, r)
- Knowing (x, r) allows us to spend commitment using the “ritual”.
- Use (x, r) as a “private key”.

BLOCK MERGING

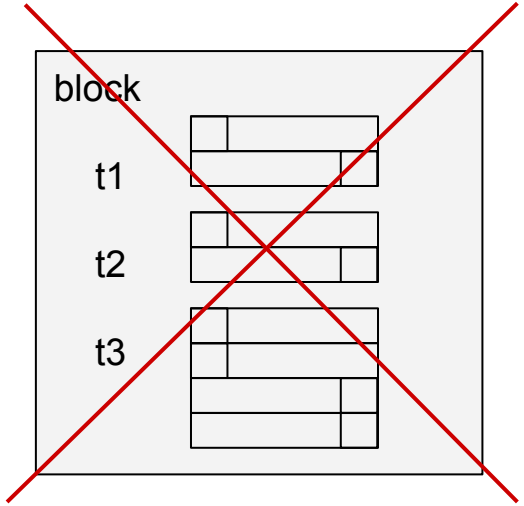
CT-BASED ANONYMIZATION

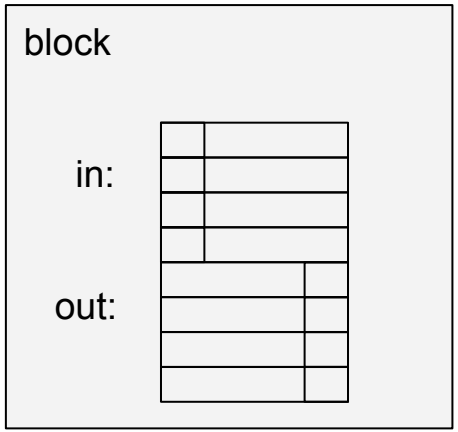
Mimblewimble Cool Idea:

- Once a commitment is spent, it is useless.
- We also don't need addresses anymore.
- Can we save lots of space by exploiting this new information?

Block Merging:

- View block not as collection of transactions, but as a single transaction, with lots of inputs and outputs.
- Like a **non-interactive CoinJoin** across all transactions in a block!





AUTHOR: PHILIP HAYES

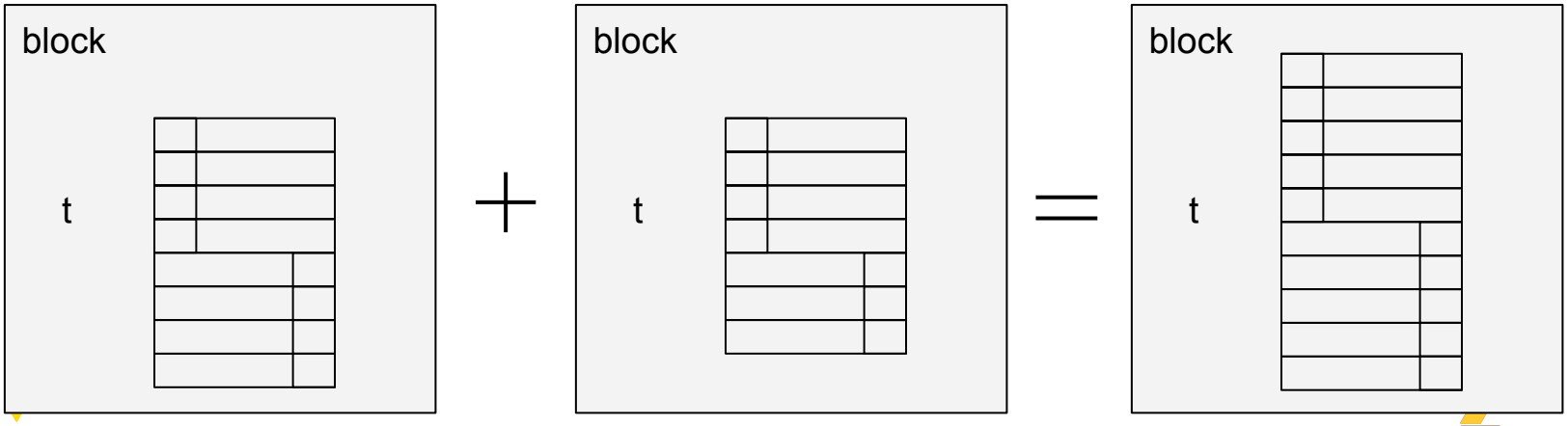
BLOCK MERGING

CT-BASED ANONYMIZATION

Idea:

- What if we just combine all of these block “transactions” into a single, giant block “transaction”?

- All of the intermediate outputs are simply eliminated in the block merging.
- To validate block, nodes just add up all the inputs and outputs and verify that it sums to zero.



MIMBLEWIMBLE

CT-BASED ANONYMIZATION

- With block merging, we just merge all the blocks back to the genesis block.
- Now, we're just left with the UTXOs (Unspent Transaction Outputs)
- Big space savings
  - Storing all blocks in current blockchain
    - ~200 GB
  - Storing equivalent Mimblewimble UTXO set
    - ~60 GB
- Anonymity!
- Actual implementation called Grin project:
  https://github.com/ignopeverell/grin

Conclusion

Rough comparative level of anonymity:
(least anonymous to most anonymous)

1. Bitcoin
2. Centralized mixers
3. Decentralized mixing protocols
4. Altcoin exchange
5. DASH
6. Confidential Transactions
7. Monero
8. Mimblewimble
9. Zcash

Practical question: **How would I mix coins today?** (In March 2017)

- Probably altcoin exchange through DASH/Monero/Zcash + TOR/VPN + throwaway exchange accounts and emails + multiple exchanges

Not covered in this lecture:

- Lightning Network and Onion Routing

Thank You!

Contact me!

- Website: maxfa.ng
- Email: max@blockchain.berkeley.edu