Decentralized Systems

Anonimity

Financial Privacy—the Good

- Which **goods one buys**
- **Supply chains** for businesses
- Financial **status**
  - Criminals target wealthy people
- Protecting against **oppressive governments**

Financial Privacy—the Bad

- Tax evasion
- Money laundering
- Criminal activities
- Financing sanctioned governments
- Unfortunately, very often, the good and the bad are **technically undistinguishable**

References

- Narayanan et al. Bitcoin and Cryptocurrency Technologies, Chapter 6 (free draft version).
- Narayanan et al. BTC-Tech: Bitcoin and Cryptocurrency Technologies. Princeton University online course, lecture 6 (also on Coursera).
- Andrei Savdeiev. Monero explanatory videos.

Anonymity

Pseudonimity and Anonymity

- Literally, “anonymous” = **without a name**
- We have seen **public key hashes,** not real names
  - In computer science, we call this **pseudonimity**
  - You can have as many pseudonyms as you want
- **Unlinkability**: different actions of the same user shouldn’t be **linkable to each other**
- **Anonymity = pseudonimity + unlinkability**

Is Unlinkability Needed?

- Pseudonimity can be **fragile**
- Many cryptocurrency services require **real identities**
  - Know-Your-Consumer (KYC)
  - People interacting with you can get personal information
- Side channels, based on **extra information** leaked
  - E.g., you send payments when you're awake and online, and you also post on social media in the same periods
  - In the long run, this may uncover who you are

Unlinkability in Cryptocurrencies

- It should be **hard** to link
  - Different **addresses** of the same user
  - Different **transactions** of the same user
  - The **sender** of a payment/message to its **recipient**

Anonymity Set

- The **crowd** one is **hiding into**
- We need to define an **adversary model**
- What they
  - **Know**
  - **Don’t know**
  - **Can’t know**

So, Bitcoin and Ethereum?

- Transactions are pseudonymous
  - They are **public forever**
  - We'll see the problem of **linkability**
- Privacy bottleneck in **exchanges**
  - **Converting** cryptocurrency to **fiat currency** (€, $)
  - And vice versa

Deanonymizing Bitcoin

Unlinkability

- Best practice: **always receive payments at a fresh address**
- Does this choice guarantee unlinkability?
- **Not necessarily**. This helps, but we can recover **patterns to link addresses**
  - When receiving (time, price, …)
  - When spending

Alice Buys a Teapot

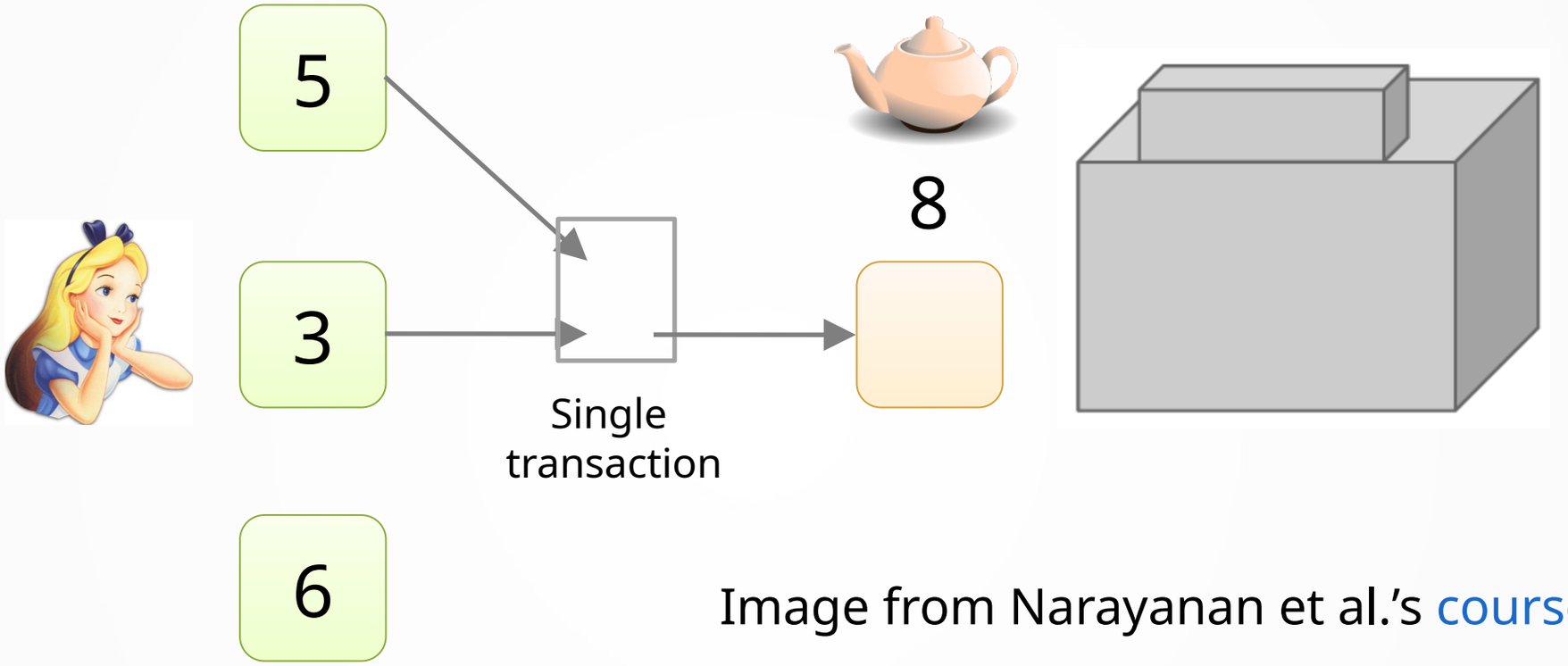


Image from Narayanan et al.'s course

- **Shared spending** may indicate **joint control** (i.e., the **same owner**)
- Addresses can be linked **transitively**

Alice Needs Change

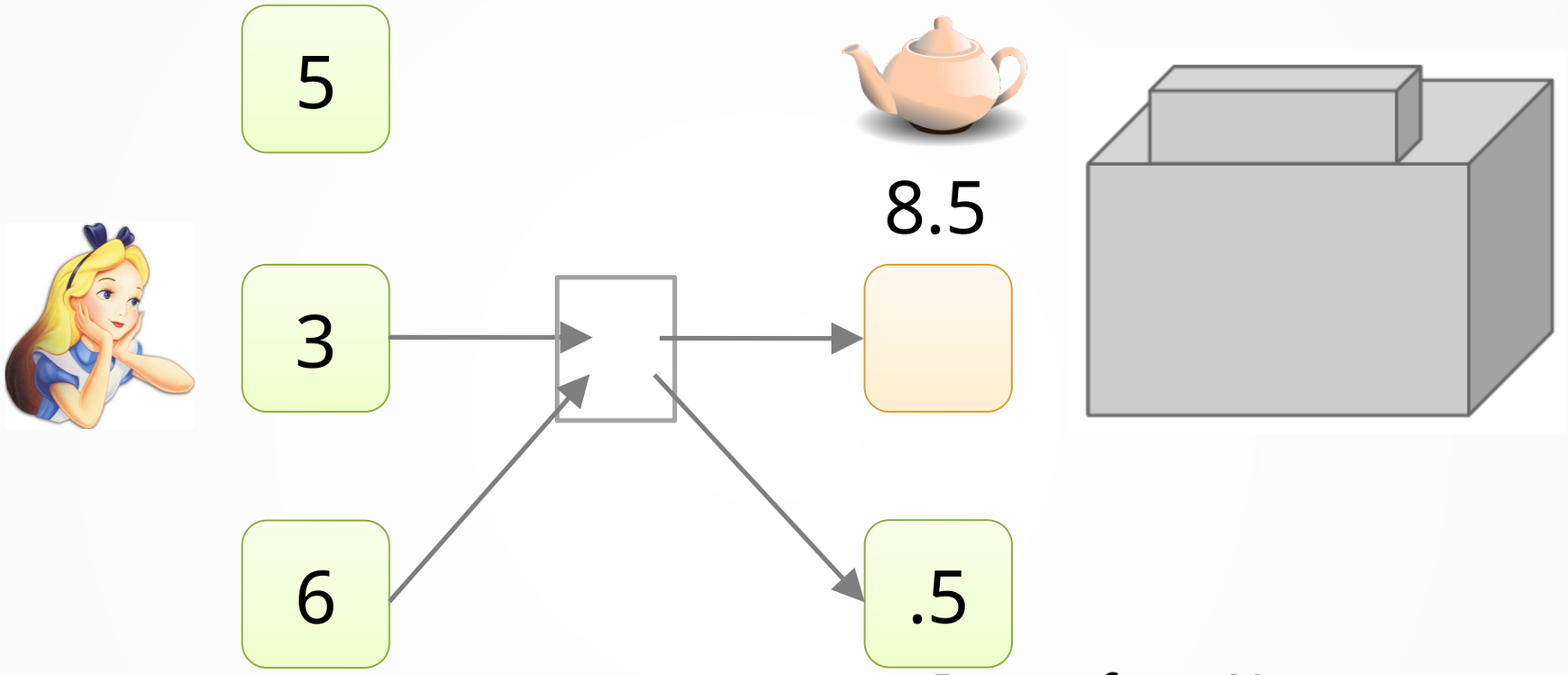


Image from Narayanan et al.'s course

- How to determine **which address is change?**
- In this case, it wouldn't make sense to combine two inputs if the cost is 0.5
- Other features depend on **how wallets are programmed**
  - E.g., last output in the transaction

Transaction Graph Analysis (1)

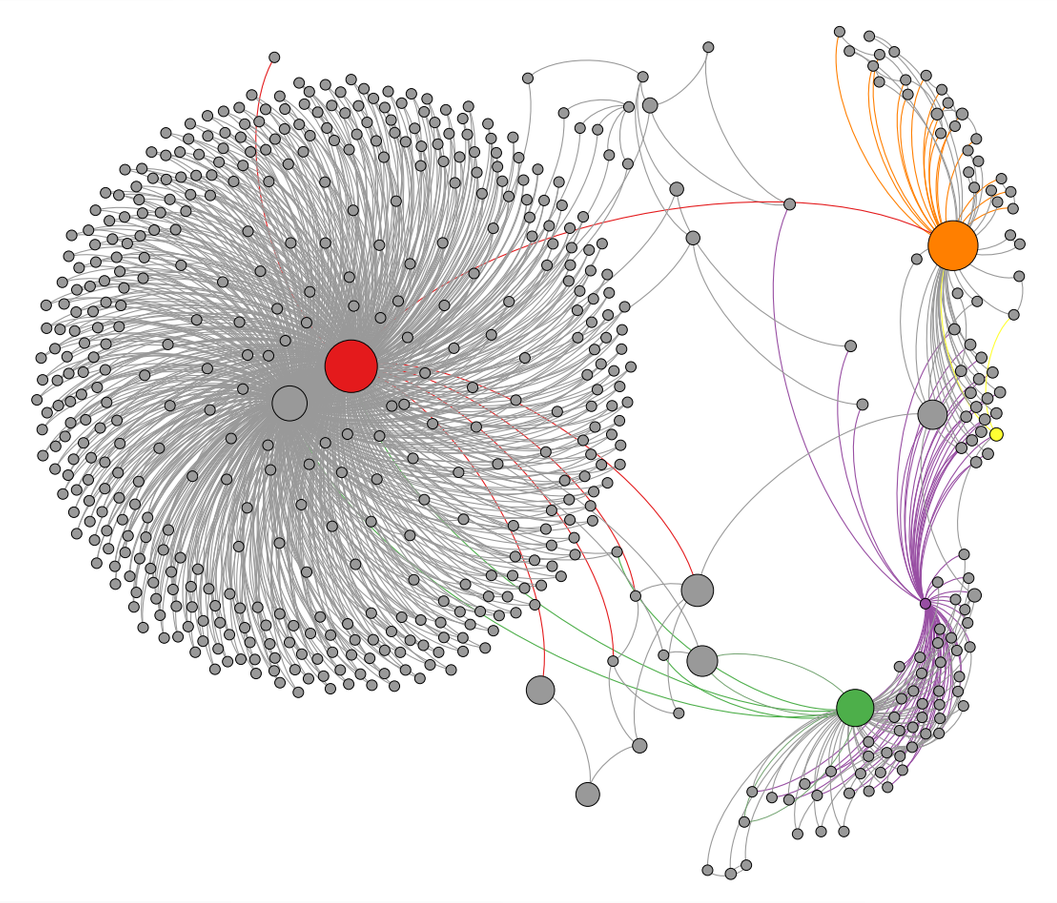


Image from Reid and Harrigan 2012

- Associate addresses that spent currency together
- In this cases, if some nodes (colored) are identified you can spot histories of payments
- This paper checks the story of Bitcoin coming from a theft

Transaction Graph Analysis (2)

- Nodes are **clusters of Bitcoin addresses**, size proportional to the transaction volumes
- Edges are **transactions**
- Authors deanonymized clusters by **transacting** with actors

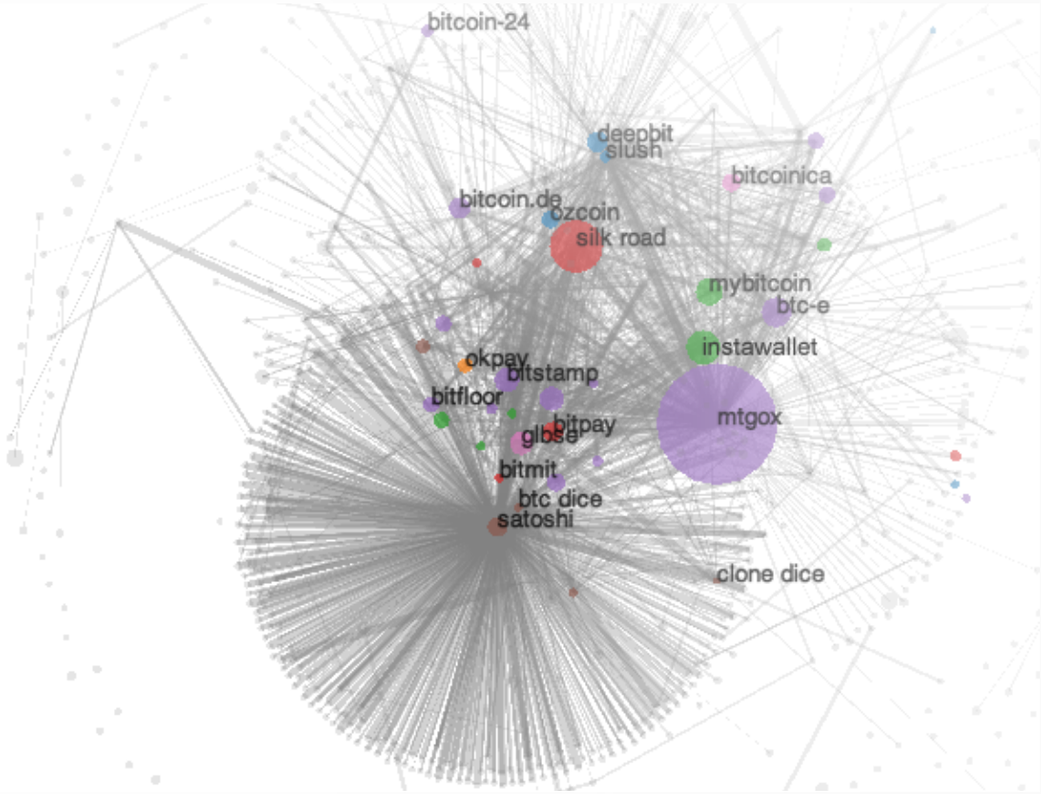


Image from Meiklejohn et al., IMC 2013

Deanonymizing Regular Users

- If one interacts with **popular services**
  - Authorities can **subpoena** them
  - Malicious entities can **attack** or **corrupt** them
- If one **publicly posts** one of their addresses
  - They can be linked with the others

Network Layer Deanonymization

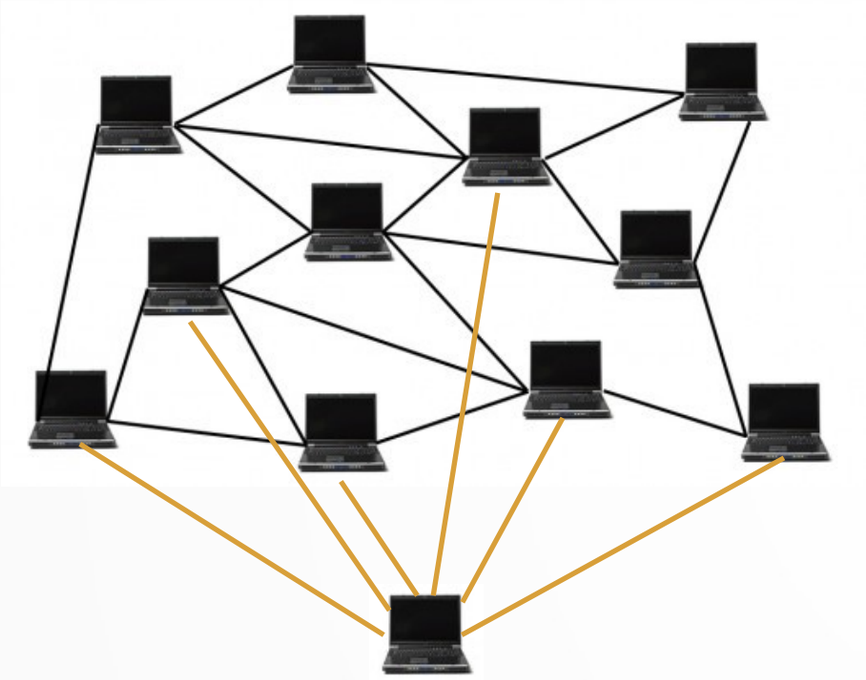


Image from Narayanan et al.'s course

- Connect to **as many machines as possible** in the Bitcoin network
- Way easier than an Eclipse attack
- "The first node to inform you of a transaction is probably the source of it" (Dan Kaminsky, BlackHat 2011 talk (slides))
- Countermeasure: use Tor or similar software

Mixing

Centralized Mixers

- Services that **receive money from a given set of addresses** and return them to **other addresses**
- Different from exchanges in that they **promise not to record identities**
- Rely on **trust** and **reputation**

Decentralized Mixing: CoinJoin

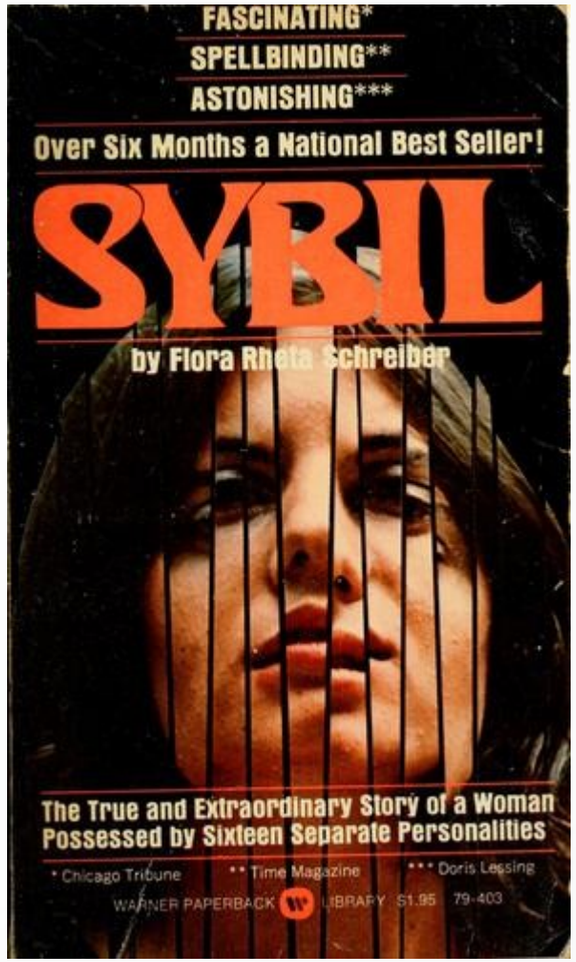
- Users find each other and build a transaction that sends money to **fresh addresses**
- They can sign if they see that it sends “their” money to the right address
- Anybody can send the tx

From	To
1 coin address 17	1 coin address 33
1 coin address 42	1 coin address 67
1 coin address 73	1 coin address 73

- This is **one** mixing round
- One generally wants to use **multiple ones** to increase anonymity set size

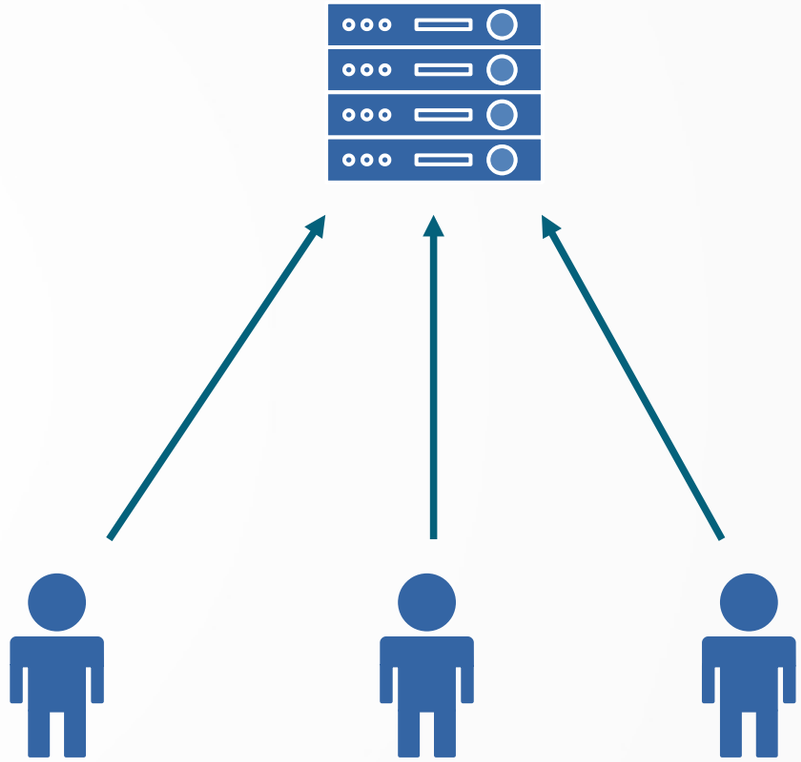
CoinJoin: Problems

- How to **find peers**
- Peers know the **mapping between inputs and outputs**
  - With a Sybil attack, you can learn it even on **multiple rounds**
- **Denial of Service**
  - A node disappears before signing
  - A node double spends the input before it passes to CoinJoin



CoinJoin: Finding Peers

- Easy! Just use an **untrusted server**
- If you think about it, the worst thing the server can do is **stop working**



CoinJoin: Mapping Inputs and Outputs

- Nodes can **use Tor** (or another anonymity solution)
- They need to use **different circuits** when communicating the inputs and the outputs
  - That way, they should be unlinkable

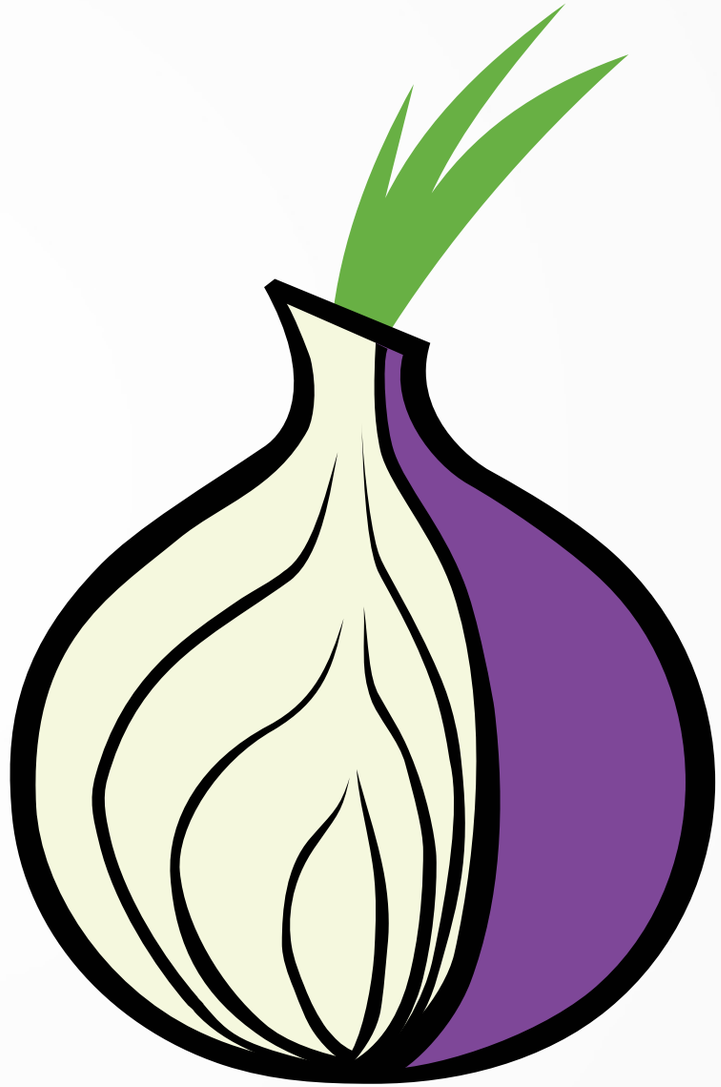


Image extracted from the Tor logo

CoinJoin: Denial of Service

- **Proof of work**
  - You must compute some hashes to talk to peers
- Proof of **burn**
  - You must destroy a small amount of currency (e.g. send to unspendable address)
- There are **cryptographic alternatives** that allow kicking non-cooperating users without revealing them

High-Level Flows

- Say Alice gets a weekly salary of 127.1425152 coins
- She puts 10% of it in a savings account right away
- This is a pattern that can be noticed **no matter what**

Merge Avoidance

Rather than a single transaction

- The receiver provides **multiple output addresses**
- The sender **avoids combining** different inputs

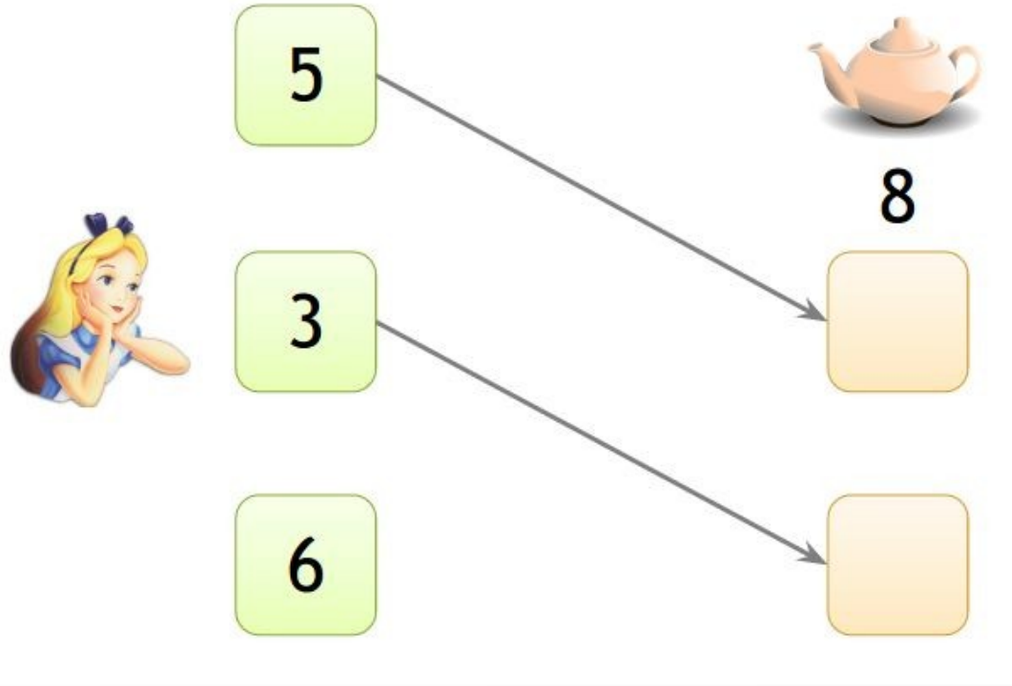


Image from the Princeton text book

Monero

Monero

- Esperanto for “coin”
- A proof-of-work cryptocurrency (XMR) designed for **anonymity** and **fungibility**
- **Fungible** goods are **interchangeable**
  - One may not accept bitcoins that are **tainted** because they come from theft, or a mix
  - Hence, Bitcoins may be **not fungible**
- Unsurprisingly, **liked by criminals**

Keys

- Each user has two asymmetric keypairs: **view** and **send**, which are not published on the blockchain
- To send XMR to Bob, Alice has to obtain **both public keys** of the recipient
- Bob’s private **view key** allows **reading** all transactions sent to Bob
- Bob’s private **send key** allows **spending** his XMR

Stealth Addresses

- With Bob’s public and view keys plus some random data, Alice generates a **stealth address** for Bob
- Blockchain transactions are sent to this **stealth address**
  - She can later **prove** she **sent money to Bob**
  - The stealth address is **unlinkable** to Bob
- Bob scans the whole blockchain using his view key to **find which transactions are for him**
  - To spend them, he can compute a **one-time secret** for each of them to spend them that will be used **together with his spend key**

Ring Signatures

- Originally called **group signatures** (Rivest et al. 2001)
- Meaning: “this document has been signed by X, Y or Z”
  - You can’t know who among them, though
- “This transaction is using funds from one output among A, B, C or D”
- How to prevent double spend? Using a **key image**
  - Unique crypto key derived from an output (and the send key)
  - Miners check it’s never reused

Ring Confidential Transactions

- Hide the **amount** of the transaction
  - Before 2017, transactions could only have fixed amounts
  - Think cash: only 1, 2, 5, 10, 20, 50 euro notes…
- Old or newly-minted XMR need to be converted to RingCT outputs
- Miners verify a crypto proof that
  - The **sum of inputs** is **equal to the sum of outputs**
  - Every output is larger than zero

In Summary

- For a Monero transaction,
  - Ring signatures hide the **sender**
  - Ring confidential transactions hide the **amount**
  - Stealth addresses hide the **recipient**
- Moreover, Kovri (a Tor-like system) hides IP addresses