Privacy for Cryptocurrencies

What information might a user want to hide?

Identity (anonymity):
- Who they are
- Who they pay
- Who pays them

Amounts:
- How much they are paying
- How much are they receiving
- E.g. salary

Metadata:
- Script Sig, e.g. multisig threshold
- Smart contract
Anonymity

Weak Anonymity (Pseudonymity):
One consistent Pseudonym (e.g. reddit)
Pros: Reputation
Cons: Linkable posts, one post linked to you -> all posts linked to you
Writing style, topics of interest may link you

Strong Anonymity:
Cons: No Reputation
Who needs privacy for payments

Companies:

- Ford does not want to reveal cost of tires
- Salaries of employees
- Investment funds want to keep strategies private
Who needs privacy for payments

Consumers

• Salary, Rent, Purchasing things online, Donations
Who needs privacy for payments

Criminals:

- Stolen funds (WannaCry), buying/selling drugs, tax evasion
Who needs privacy for payments

Applications:

• Privacy can prevent frontrunning
• Exchanges may want to keep orderbook private
• Sealed bid auction
Privacy of Digital Payments

Payments publicly visible/linkable

Payments only visible to bank/venmo. Optionally sender/receiver public

Unlinkable private payments

Less private  More private
Privacy in Ethereum

Weak Pseudonymity:
- Account public
- Values public
- Mostly one account per user
- Some accounts known (Binance)
# Privacy in Bitcoin

## Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>1110 (bytes)</td>
</tr>
<tr>
<td>Fee Rate</td>
<td>0.0016173243243243244 BTC per kB</td>
</tr>
<tr>
<td>Received Time</td>
<td>Apr 10, 2017 12:38:00 AM</td>
</tr>
<tr>
<td>Mined Time</td>
<td>Apr 10, 2017 12:38:00 AM</td>
</tr>
<tr>
<td>Included in Block</td>
<td>00000000000000001f0115cca585646832b337404032c88539ce2995e799e5c</td>
</tr>
</tbody>
</table>

## Details

<table>
<thead>
<tr>
<th>Transaction Hash</th>
<th>Amount</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2561b292ed4878bb28478a8cafd1f99a01faeb9c5a906715fa595cac0e8d1d8</td>
<td>0.53333328 BTC</td>
<td>Input</td>
</tr>
<tr>
<td>16k4365RzdeCPKGwJDNNEkXj696MbChwx</td>
<td>0.01031593 BTC</td>
<td>Output</td>
</tr>
<tr>
<td>1Bsh4KD9ZJT4dJcoo7S5u1jvtmtVmREb7</td>
<td>1.47877788 BTC</td>
<td>Input</td>
</tr>
<tr>
<td>1JgVBpwSTDMTjRoZXg9XpPDQRHtNb5CsPA</td>
<td>2 BTC</td>
<td>Output</td>
</tr>
</tbody>
</table>

FEE: 0.00179523 BTC

---

Note: The transactions above are hypothetical examples and do not reflect real Bitcoin transactions.
Privacy in Bitcoin

Alice can have many addresses (creating address is free)

Ins: A1: 4 A2: 5  
out: B: 6, A3: 3  

Change address

Alice’s addresses  
Bob’s address
Linking Addresses to Identities

- Buying book from merchant
  - Alice learns one of merchant’s addresses (B)
  - Merchant learns three of Alice’s addresses
- Alice uses an exchange \(\text{BTC} \leftrightarrow \$\)
  - KYC (Know your customer)
- Money serving business collect and verify IDs
Linking Addresses to Identities

- Buying book from merchant
  - Alice learns one of merchant’s addresses (B)
  - Merchant learns three of Alice’s addresses
- Alice uses an exchange $BTC \leftrightarrow \$$
  - KYC (Know your customer)
  - Money serving business collect and verify IDs
  - Exchange learns real ID

*(Ins: A1: 4 A2: 5 out: B: 6, A3: 3)*
Donating to Wikileaks

WikiLeaks had one address -> Easy to see who donates
Is Bitcoin Anonymous?

No!

It is possible to:

• Link all addresses of a single entity:
  • Determine total assets
• Given two TX A->B, C->D, Are B&C the same
  • If D knows C, can unmask B
  • Trace stolen funds, find tax evasion
  • Oppressive governments (Venezuela, North Korea)
• Test if Alice ever paid Bob (Wikileaks)

Often answer is yes for all 3. How?
Network Anonymity

end users

\( \text{sk}_A \)

\( \text{sk}_B \)

\( \text{sk}_C \)

Can learn Alice’s IP address

Bitcoin P2P network

Solution:

Tōr
Light client network anonymity

SPV client

All addresses and transactions

Full node

Fully linkable!
Idioms of use

Heuristic 1:

Two addresses are input to same TX (and not multisig script)

-> both addresses are controlled by same entity
Idioms of use

Heuristic 2:
Change address is controlled by same user as input address
Which is change address: Used to be first address
Heuristic: Only new address, Non round, Less than inputs
Example tracing

Coinbase knows entity!
Experiment (2013)

- Use Heuristic 1 and 2 -> 3.3M clusters
- ID 1070 addresses by interacting with merchants
  - Coinbase, Bitpay, ...
- Learn ID of 2200 clusters
  - 1.8M address
  - 15% of total value
- Track multiple thefts
- Learn total assets for each cluster
Making Cryptocurrencies anonymous

Mixing

Anonymous cryptocurrencies
Another example

Alice and Subcontractor learn EC’s profit margin.
How can we prevent this?
Another example

EC has many customers. Mix payments -> use some to pay sub
Mixing

A1 -> M: 1
B1 -> M: 1
C1 -> M: 1

A2
B2
C2

Mixer

TLS

Ins: M: 3 Outs: B2: 1, A2: 1, C2: 1
Mixing Analysis

• Outside observer who is A2?
  • $A_2 \in \{Alice, Bob, Carol\}$
• For Bob
  • $A_2 \in \{Alice, Bob, Carol\}$
• The more the better mixing
Mixer Problems

- Mixer can deanonymize
- All outputs MUST have same value
  - If not you can match inputs and outputs
- Mixer takes transaction fees
- Mixer can steal funds
- ScriptPK for all outputs must be the same
  - Otherwise linkable on spend
CoinJoin TX

Ins: A1: 5, B1: 3, C1: 2
Outs: B2: 2, A2: 2, C2: 2
Change (not private): A3: 3, B3: 1
Signed: Multisig A1, B1, C1

Out value = min of inputs

Usually ~40 inputs
CoinJoin

A1: 5, A3 (change)
A2 (over Tor)
Add Signatures
Publish Transaction

Online Forum

PASTEBIN

A1: 5, A3
B1: 3, B3
C1: 2, C3
B2, A2, C2

What if A1 is spent?
Coinjoin still has drawbacks:

- Interaction required
- Any party can disrupt the process
- Anonymity set determined by who is using the service
- Transaction amounts public
Cryptonote (Monero)

• Cryptonote protocol, proposed in 2012
• Enables non interactive coinjoin
• Sender can choose anonymity set
• Hides amounts
• Basis of Monero, Mobile coin, others
Def: A signature scheme is a triple of algorithms:

- **Gen()**: outputs a key pair \((pk, sk)\)
- **Sign\((sk, msg)\)** outputs sig. \(\sigma\)
- **Verify\((pk, msg, \sigma)\)** outputs ‘accept’ or ‘reject’

**Secure signatures** (informal)

Adversary who sees signatures on many messages of his choice, cannot forge a signature on a new message.
Linkable Ring Signatures

**Def**: a signature scheme is a triple of algorithms:

- **Gen()**: outputs a key pair \((pk, sk)\)
- **RingSign**(sk, PKs, msg) outputs sig. \(\sigma\)
- **Verify**(pk, PKs, msg, \(\sigma\)) outputs ‘accept’ or ‘reject’
- **Link**(PKs, msg, \(\sigma\), PKs’, msg’, \(\sigma’\)) outputs 0 or 1

**Secure signatures**: (informal)

Unforgeability: Adversary who sees signatures on many messages of his choice, cannot forge a signature on a new message.

Anonymity: \(\text{Sign}(sk_i, PKs, msg) \approx \text{Sign}(sk_j, PKs, msg)\) for \(pk_i, pk_j \in PKs\)

Linkability: If a secret key signs two messages, then the signatures can be linked
CryptoNote

**Additional Pieces:**
- Generate \( PK_s \) without interaction
- Make amounts private (next lecture)

---

**TX:** Inputs \( PKs \), Output: \( PK_R \), Signature: \( \text{Sign}(sk, PKs, TX) \)
CryptoNote analysis

• Sender picks anonymity set
  • Ring signature provides anonymity in set
  • The larger the set the better
  • Still not perfect (e.g. if I know all other PKs in set)
• Linkability of ring signatures prevents double spends
• Keys can only be used once
• Hides amounts (unlike coinjoin)
• Fully non interactive
Next lecture:
Zero-knowledge SNARKs