Privacy in Bitcoin

On the Effectiveness of Clustering

Jonas Nick

March 15, 2016

Privacy

Privacy

- Anonymity
  - "Silkroad, anonymous market" - Bitcoin drug market

Privacy

- Anonymity
  - "Silkroad, anonymous market" - Bitcoin drug market
  - "... the major advantage they [Bitcoin] are providing is anonymity."
    - NY's Department of financial services

Privacy

- Anonymity
  - "Silkroad, anonymous market" - Bitcoin drug market
  - "... the major advantage they [Bitcoin] are providing is anonymity."
    - NY's Department of financial services
  - "... usually not very anonymous." - Bitcoin wiki

Privacy

- Anonymity
  - "Silkroad, anonymous market" - Bitcoin drug market
  - "... the major advantage they [Bitcoin] are providing is anonymity."
    - NY's Department of financial services
  - "... usually not very anonymous." - Bitcoin wiki

Privacy

- Why?
  - Privacy and fungibility essential characteristics of money.
- What?
  - Anonymity + Selective Transparency

Privacy

- Why?
  - Privacy and fungibility essential characteristics of money.
- What?
  - Anonymity + Selective Transparency
- Good news: That's possible

Privacy

- Why?
  - Privacy and fungibility essential characteristics of money.
- What?
  - Anonymity + Selective Transparency
- Good news: That's possible
- This talk: There's a long road road ahead

Privacy

- Bitcoin is *pseudonymous*
- entities (persons, companies, etc.) are represented by *public keys* ($\approx$ *addresses*)
  - unbounded number of public keys per entity

Privacy

- Bitcoin is *pseudonymous*
- entities (persons, companies, etc.) are represented by *public keys* ($\approx$ *addresses*)
    - unbounded number of public keys per entity
- sender public keys, recipient public keys and values of transactions are public

Privacy

- Bitcoin is *pseudonymous*
- entities (persons, companies, etc.) are represented by *public keys* ($\approx$ *addresses*)
    - unbounded number of public keys per entity
- sender public keys, recipient public keys and values of transactions are public
- unknown which public keys belong to an entity

Privacy

- Bitcoin is *pseudonymous*
- entities (persons, companies, etc.) are represented by *public keys* ($\approx$ *addresses*)
    - unbounded number of public keys per entity
- sender public keys, recipient public keys and values of transactions are public
- unknown which public keys belong to an entity
- *Clustering*: Given public key, use blockchain to find public keys owned by the same entity.

`<Friedrich_Nietzsche>: Glad that I could help - would be great if you pass me some bitcoin 1GsYQYsgf1zmwY8LAsgEMD`

`<Friedrich_Nietzsche>: Glad that I could help - would be great if you pass me some bitcoin 1GsYQYsgf1zmwY8LAsgEMD`

In blockchain:

$\texttt{1FgtvT2W45nZi9fr3jsVRt} \xrightarrow{\text{1 bitcoin}} \texttt{1abcDogDating}$

`<Friedrich_Nietzsche>: Glad that I could help - would be great if you pass me some bitcoin 1GsYQYsgf1zmwY8LAsgEMD`

In blockchain:

$\texttt{1FgtvT2W45nZi9fr3jsVRt} \xrightarrow{\text{1 bitcoin}} \texttt{1abcDogDating}$

Clustering reveals both addresses are from same wallet

Transactions

- balance-based vs. UTXO model

Transactions

- balance-based vs. UTXO model
- balance-based (f.e. Ethereum)
  - Blockchain state

Alice	2
Bob	0

Transactions

- balance-based vs. UTXO model
- balance-based (f.e. Ethereum)
  - Blockchain state

Alice	2
Bob	0

  - Transaction: $\texttt{Alice} \xrightarrow{\text{1 coin}} \texttt{Bob}$

Transactions

- balance-based vs. UTXO model
- balance-based (f.e. Ethereum)
  - Blockchain state

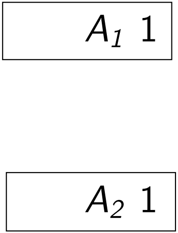
Alice	2
Bob	0

  - Transaction: `Alice` $\xrightarrow{\text{1 coin}}$ `Bob`
  - new Blockchain state

Alice	1
Bob	1

Transactions

- UTXOs (Unspent Transaction Outputs)
- Bitcoin's model



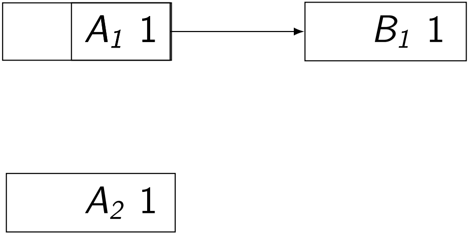
A_1 1

A_2 1

- Balance implicit
- Cash analogy

Transactions

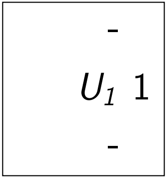
- UTXOs (Unspent Transaction Outputs)
- Bitcoin's model



- Balance implicit
- Cash analogy

Transactions

Transactions



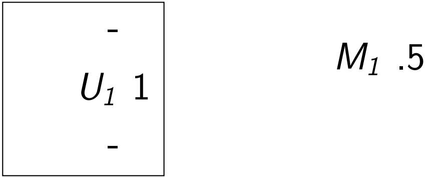
-

U_1 1

-

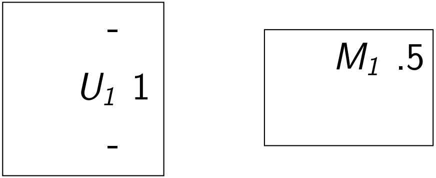
- user U, merchant M
- spend tx *outputs* (value and recipient)

Transactions



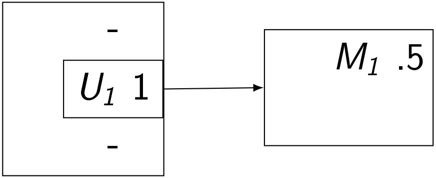
- user *U*, merchant *M*
- spend tx *outputs* (value and recipient)

Transactions



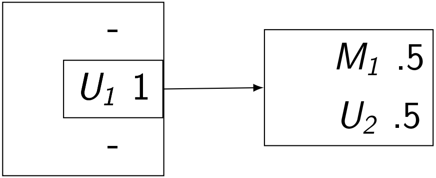
- user U, merchant M
- spend tx *outputs* (value and recipient)

Transactions



- user U, merchant M
- spend tx *outputs* (value and recipient)
- *inputs*

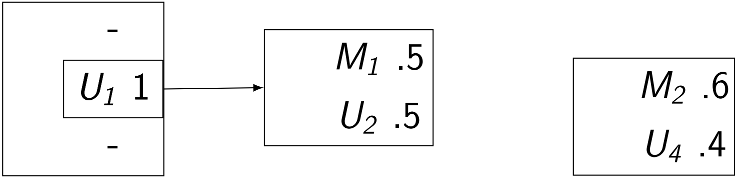
Transactions



- user U, merchant M
- spend tx *outputs* (value and recipient)
- *inputs*

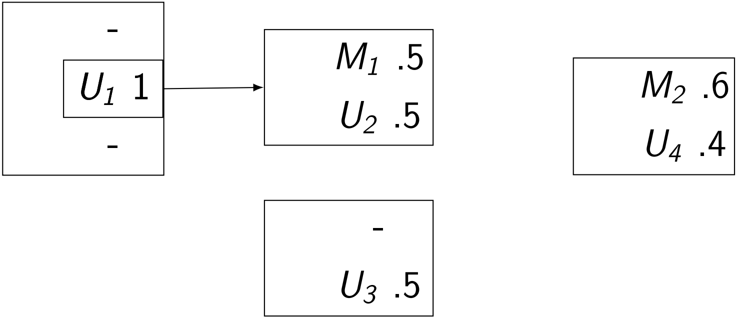
- *change*

Transactions



- user U, merchant M
- spend tx *outputs* (value and recipient)
- *inputs*
- *change*

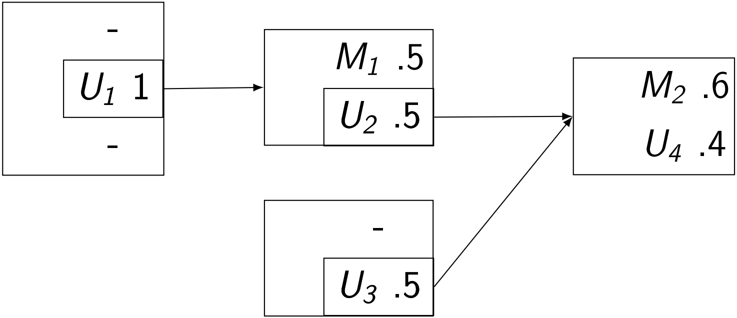
Transactions



- user U, merchant M
- spend tx *outputs* (value and recipient)
- *inputs*

- *change*

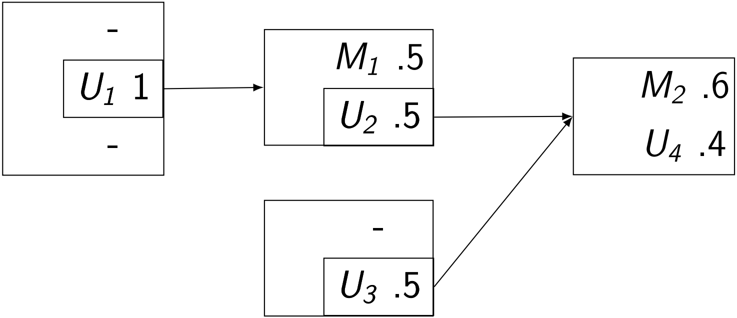
Transactions



- user U, merchant M
- spend tx *outputs* (value and recipient)
- *inputs*

- *change*
- *multi-input tx*

Transactions



- user U, merchant M
- spend tx *outputs* (value and recipient)
- *inputs*

- *change*
- *multi-input tx*
- *pay-to-pubkey-hash*

Questions?

Clustering Strategies

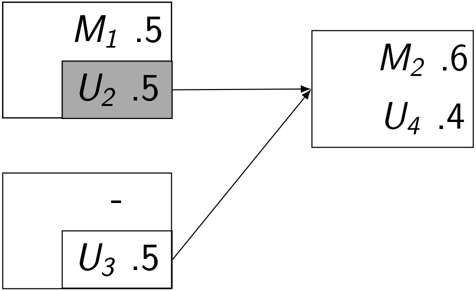
- Given pubkey, use blockchain to find pubkeys of the same wallet
- make assumptions about wallet behavior
    - heuristics

Multi-input heuristic

All inputs of a transaction belong to the same wallet.

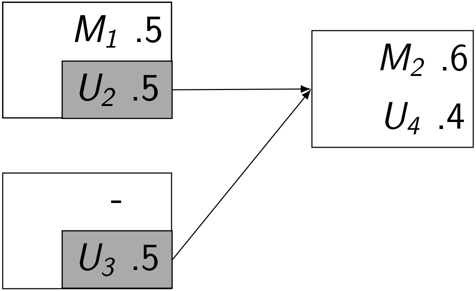
Multi-input heuristic

All inputs of a transaction belong to the same wallet.



Multi-input heuristic

All inputs of a transaction belong to the same wallet.

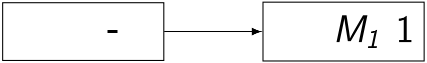


Shadow change heuristic

Change pubkeys have never been seen before in the blockchain.

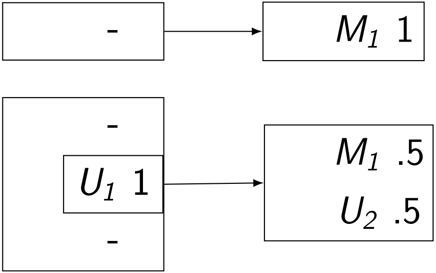
Shadow change heuristic

Change pubkeys have never been seen before in the blockchain.



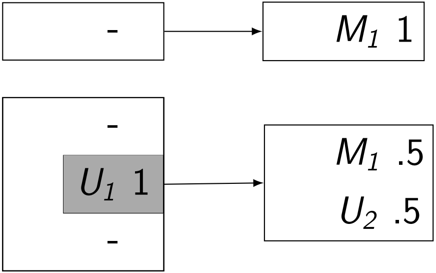
Shadow change heuristic

Change pubkeys have never been seen before in the blockchain.



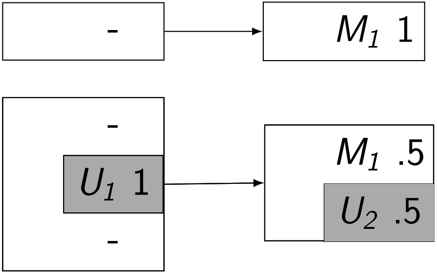
Shadow change heuristic

Change pubkeys have never been seen before in the blockchain.



Shadow change heuristic

Change pubkeys have never been seen before in the blockchain.

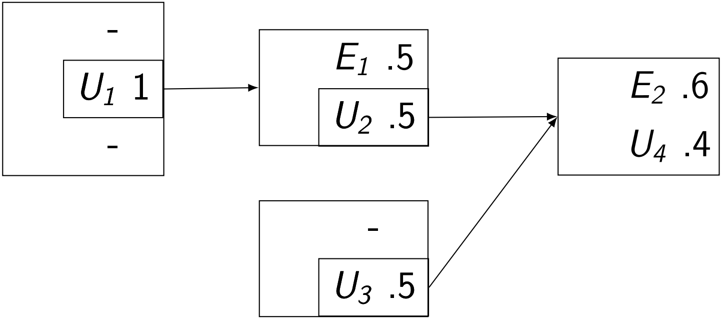


Consumer change heuristic

Transactions from consumer wallets have two or less outputs.

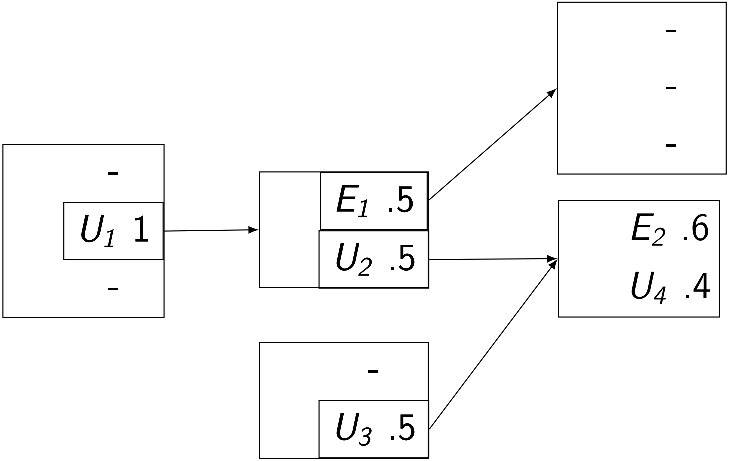
Consumer change heuristic

Transactions from consumer wallets have two or less outputs.



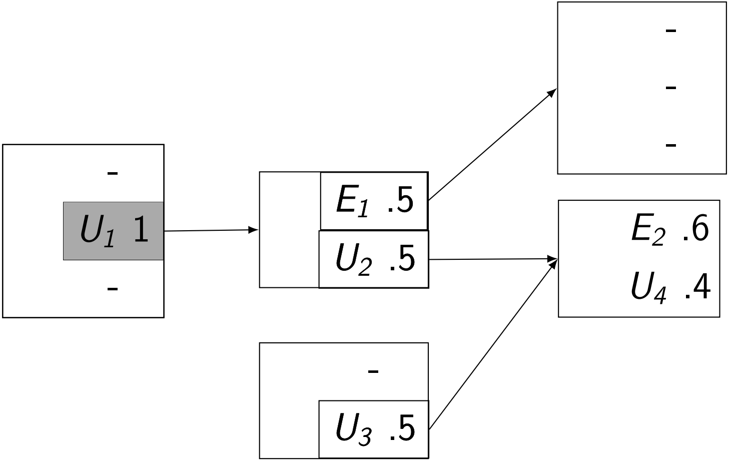
Consumer change heuristic

Transactions from consumer wallets have two or less outputs.



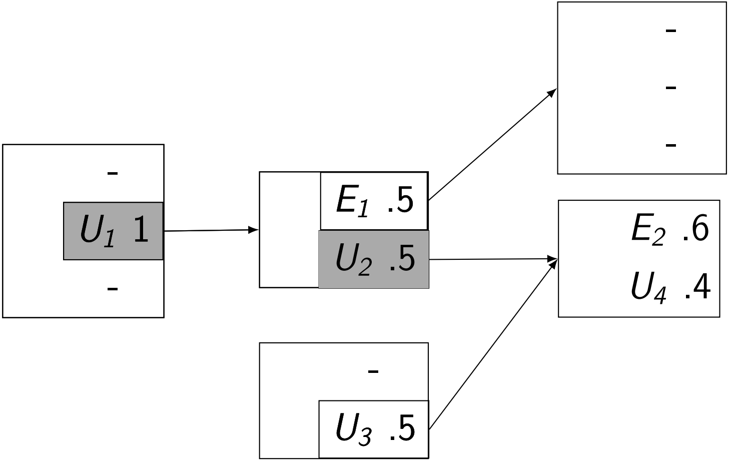
Consumer change heuristic

Transactions from consumer wallets have two or less outputs.



Consumer change heuristic

Transactions from consumer wallets have two or less outputs.

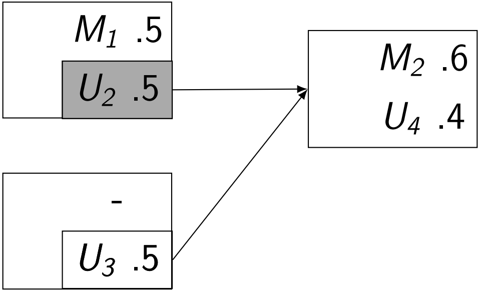


Optimal change heuristic

Wallets do not spend unnecessary outputs.

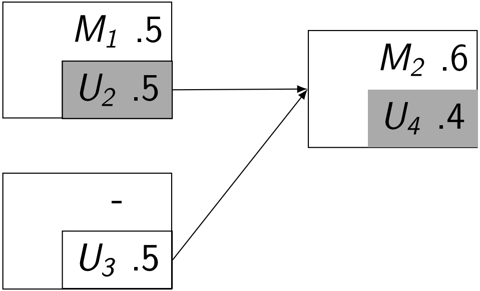
Optimal change heuristic

Wallets do not spend unnecessary outputs.



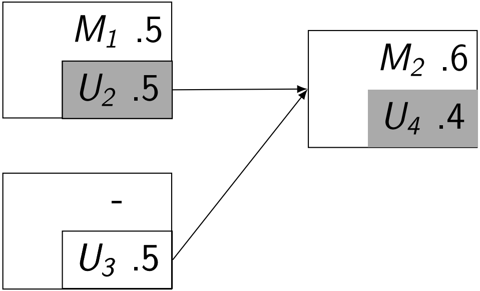
Optimal change heuristic

Wallets do not spend unnecessary outputs.



Optimal change heuristic

Wallets do not spend unnecessary outputs.



If there is a unique output with a value smaller than any of the inputs, then this is the change.

Next steps

- How to quantify privacy on the blockchain?
- Requires data...

P2P wallet leak

- *simplified payment verification* (SPV): light wallets

P2P wallet leak

- *simplified payment verification* (SPV): light wallets
- Some SPV wallets implement BIP37: *Connection Bloom filtering*
  - used for learning about new transactions concerning the wallet

P2P wallet leak

- *simplified payment verification* (SPV): light wallets
- Some SPV wallets implement BIP37: *Connection Bloom filtering*
    - used for learning about new transactions concerning the wallet
- Examples: Android Bitcoin Wallet, MultiBit, Breadwallet, etc.

P2P wallet leak

- *simplified payment verification* (SPV): light wallets
- Some SPV wallets implement BIP37: *Connection Bloom filtering*
  - used for learning about new transactions concerning the wallet
- Examples: Android Bitcoin Wallet, MultiBit, Breadwallet, etc.

Bloom Filter

Bloom Filter

- Purpose: efficiently testing if element is contained in a set

Bloom Filter

- Purpose: efficiently testing if element is contained in a set
- Operations: `insert` and `query`

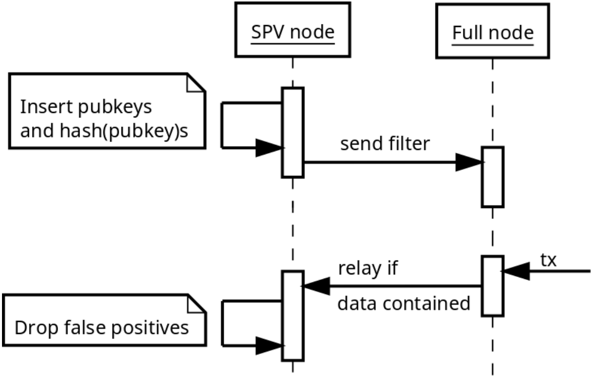
Bloom Filter

- Purpose: efficiently testing if element is contained in a set
- Operations: `insert` and `query`
- False positive rate: $\Pr(\text{query} \mid \text{not inserted})$

Bloom Filter

- Purpose: efficiently testing if element is contained in a set
- Operations: `insert` and `query`
- False positive rate: $\Pr(\text{query} | \text{not inserted})$
- No false negatives

Bloom Filter in Bitcoin



Bloom Filter in Bitcoin

- Filter has space and time advantage

Bloom Filter in Bitcoin

- Filter has space and time advantage
- fp-rate: bandwidth/privacy trade-off

Bloom Filter in Bitcoin

- Filter has space and time advantage
- fp-rate: bandwidth/privacy trade-off
- Most wallets: 8000 false positives

Bloom Filter Vulnerability

- Idea: query both pubkey and hash(pubkey)
- then Pr(query(pk and pkh)|not inserted(pk and pkh))

Bloom Filter Vulnerability

- Idea: query both pubkey and hash(pubkey)
- then $\Pr(\text{query(pk and pkh)}|\text{not inserted(pk and pkh)})$
  - $= \text{fp-rate}^2$

Bloom Filter Vulnerability

- Idea: query both pubkey and hash(pubkey)
- then $\Pr(\text{query(pk and pkh)} | \text{not inserted(pk and pkh)})$
    - $= \text{fp-rate}^2$
- most wallets: 1 false positive

Bloom Filter Vulnerability

- Idea: query both pubkey and hash(pubkey)
- then $\Pr(\text{query(pk and pkh)}|\text{not inserted(pk and pkh)})$
  - $=$ fp-rate^2
- most wallets: 1 false positive
- 20 crawlers collected 37,585 filters
- need to be picked up by seed nodes

Results

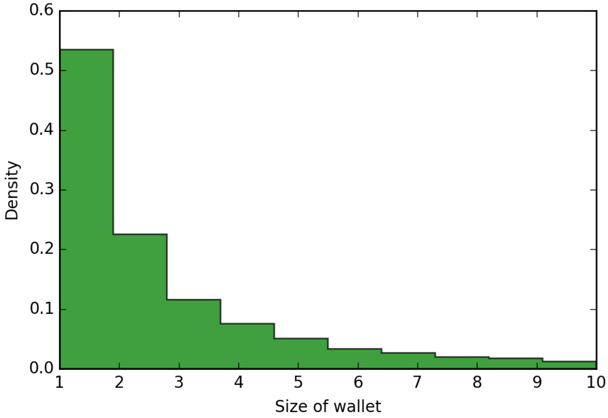


Figure Distribution of the number of pubkeys in captured BIP37 wallets.

Results

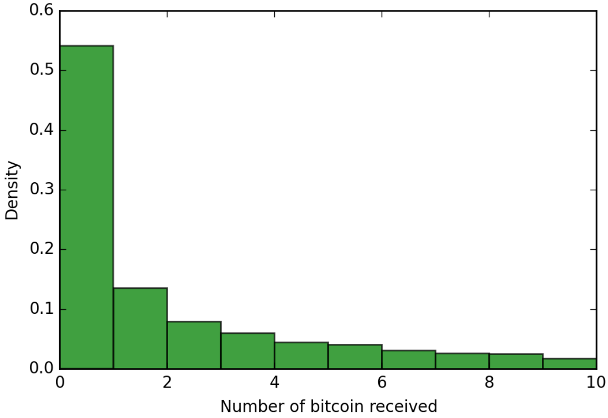


Figure Distribution of total received bitcoins for a subset of wallets.

Mitigation

- A general fix requires substantial modification of the protocol and is not on the priority list.

Mitigation

- A general fix requires substantial modification of the protocol and is not on the priority list.
- Alternatives
  - a central server that learns all of the client's addresses
  - full node

Evaluate Clustering

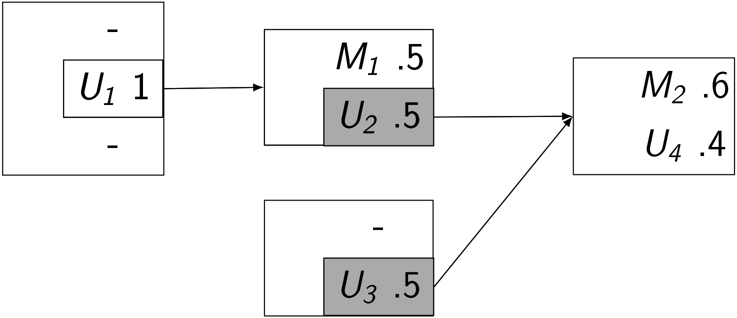
- Collected filters allow to reconstruct all pubkeys of a wallet
- Can apply clustering and evaluate clustering performance using ”Ground truth”

Performance metric

- precision: $\Pr(\text{in wallet}|\text{heuristic})$
- recall: $\Pr(\text{heuristic}|\text{in wallet})$

Performance metric

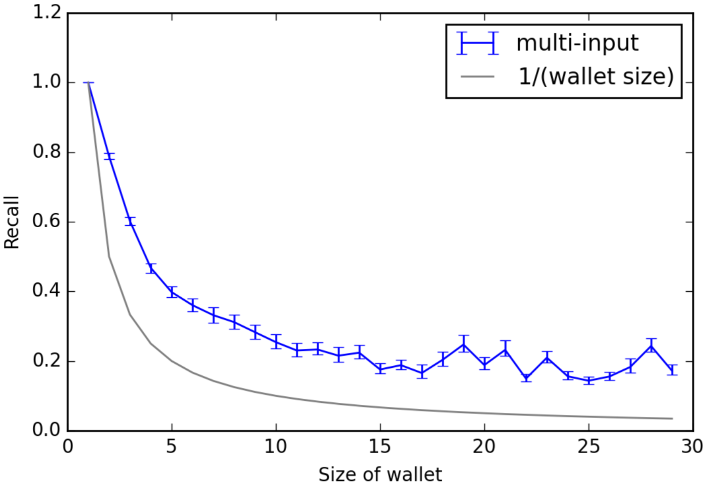
- precision: Pr(in wallet|heuristic)
- recall: Pr(heuristic|in wallet)



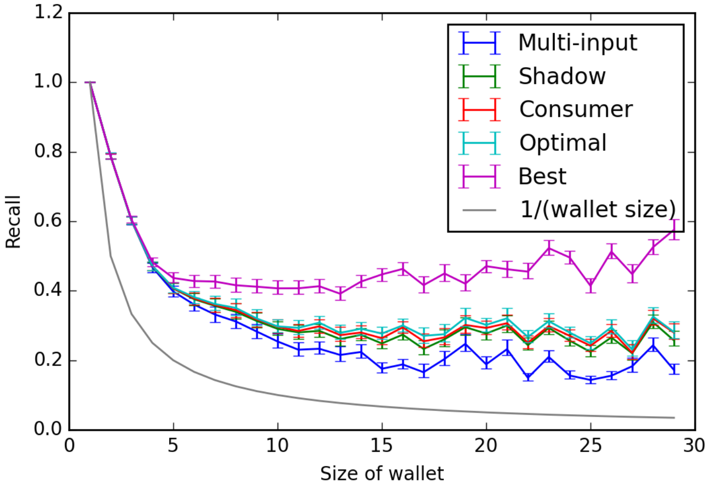
precision: 1, recall: $\frac{2}{4}$

Results

Heuristic	mean recall
1/(wallet size)	66.27%
Multi-input	68.59%
Shadow	69.16%
Consumer	69.26%
Optimal	69.34%
Best	70.94%


multi-input
1/(wallet size)
Recall
Size of wallet
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25
30

Result


Recall
Size of wallet
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
Multi-input
Shadow
Consumer
Optimal
Best
1/(wallet size)

Conclusion

- captured pubkeys of $37,000$ wallets from the Bitcoin network

Conclusion

- captured pubkeys of $37,000$ wallets from the Bitcoin network
- introduced two new clustering strategies

Conclusion

- captured pubkeys of $37,000$ wallets from the Bitcoin network
- introduced two new clustering strategies
- evaluated performance of clustering using ground truth

Conclusion

- captured pubkeys of $37,000$ wallets from the Bitcoin network
- introduced two new clustering strategies
- evaluated performance of clustering using ground truth
- modern wallets: 70% recall

Countermeasures for User

- keep your wallet up to date

Countermeasures for User

- keep your wallet up to date
- do not use wallets with Bloom filtering

Countermeasures for User

- keep your wallet up to date
- do not use wallets with Bloom filtering
- do not reuse addresses

Countermeasures for User

- keep your wallet up to date
- do not use wallets with Bloom filtering
- do not reuse addresses
- other parties play a role

Countermeasures for User

- keep your wallet up to date
- do not use wallets with Bloom filtering
- do not reuse addresses
- other parties play a role
- separate

Countermeasures for User

- keep your wallet up to date
- do not use wallets with Bloom filtering
- do not reuse addresses
- other parties play a role
- separate
- openbitcoinprivacyproject.org

Countermeasures for User

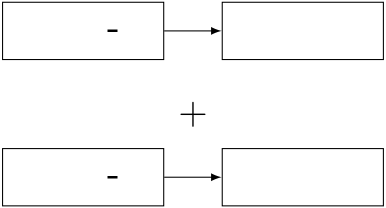
- keep your wallet up to date
- do not use wallets with Bloom filtering
- do not reuse addresses
- other parties play a role
- separate
- openbitcoinprivacyproject.org
- altcoins?

Countermeasures for Developer

- coin selection

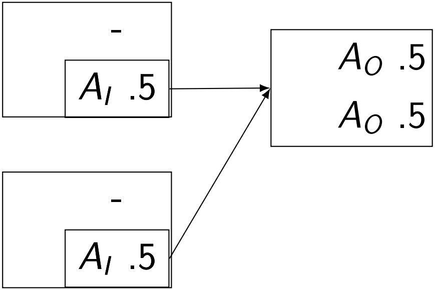
Countermeasures for Developer

- coin selection
- coinjoin



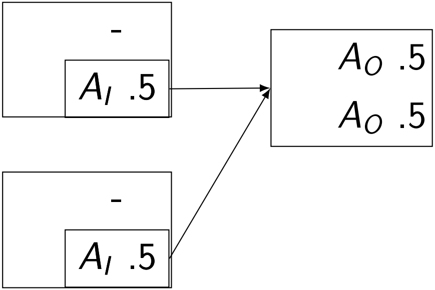
Countermeasures for User

- coinjoin



Countermeasures for User

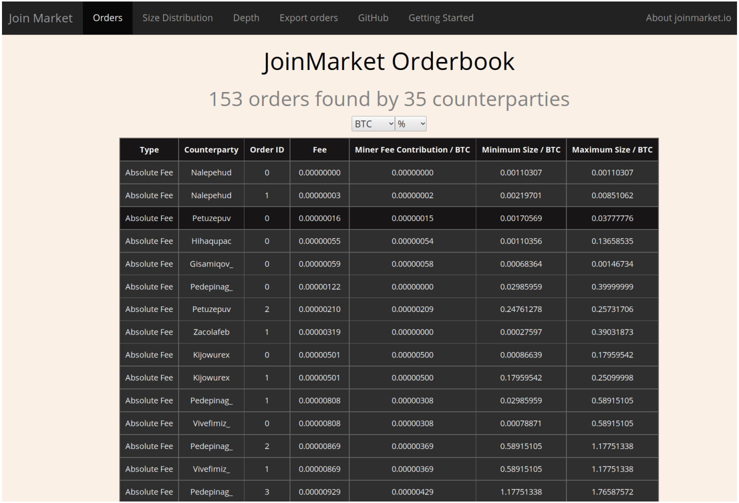
- coinjoin



  - trustless, but
  - UI, exact protocol challenging
  - Confidential transactions

Countermeasures for User

- Joinmarket



JoinMarket Orderbook

153 orders found by 35 counterparties

BTC %

Type	Counterparty	Order ID	Fee	Miner Fee Contribution / BTC	Minimum Size / BTC	Maximum Size / BTC
Absolute Fee	Nalepehud	0	0.00000000	0.00000000	0.00110307	0.00110307
Absolute Fee	Nalepehud	1	0.00000003	0.00000002	0.00219701	0.00851062
Absolute Fee	Petuzepuv	0	0.00000016	0.00000015	0.00170569	0.03777776
Absolute Fee	Hihaqupac	0	0.00000055	0.00000054	0.00110356	0.13658535
Absolute Fee	Gisamiqov_	0	0.00000059	0.00000058	0.00068364	0.00146734
Absolute Fee	Pedepinag_	0	0.00000122	0.00000000	0.02985959	0.39999999
Absolute Fee	Petuzepuv	2	0.00000210	0.00000209	0.24761278	0.25731706
Absolute Fee	Zacolafeb	1	0.00000319	0.00000000	0.00027597	0.39031873
Absolute Fee	Kijowurex	0	0.00000501	0.00000500	0.00086639	0.17959542
Absolute Fee	Kijowurex	1	0.00000501	0.00000500	0.17959542	0.25099998
Absolute Fee	Pedepinag_	1	0.00000808	0.00000308	0.02985959	0.58915105
Absolute Fee	Vivefimiz_	0	0.00000808	0.00000308	0.00078871	0.58915105
Absolute Fee	Pedepinag_	2	0.00000869	0.00000369	0.58915105	1.17751338
Absolute Fee	Vivefimiz_	1	0.00000869	0.00000369	0.58915105	1.17751338
Absolute Fee	Pedepinag_	3	0.00000929	0.00000429	1.17751338	1.76587572

Q&A

- Questions?
- Contact
  - nickler.ninja
    - slides: nickler.ninja/slides/2016-zurich-meetup.pdf
    - thesis: nickler.ninja/papers/thesis.pdf
  - jonas@blockstream.com