Transaction clustering using network traffic analysis for Bitcoin and derived blockchains

Alex Biryukov, **Sergei Tikhomirov**

SnT, University of Luxembourg

29 April 2019
Cryblock
Paris, France

UNIVERSITÉ DU LUXEMBOURG

Outline

Outline

Privacy in cryptocurrencies

- Transactions not linked to "real-world" identity
- False sense of privacy: blockchain can be analyzed
- Taint analysis, various heuristics
- Countermeasures: mixing, cryptography (Monero, Zcash, ...)

Our focus: network-level privacy

- How do messages propagate through the network?
- What information does the traffic leak?
- Is it possible to link txs by the same user?

Outline

Transaction propagation in Bitcoin

- Alice: `INV` (I know an object with hash H)
- Bob: `GETDATA` (I want to get this object)
- Alice: `TX` (Here it is)

Bob announces to his neighbors, etc.

Broadcast randomization

Privacy issue: well-connected adversary infers the original IP.
Countermeasures:

- trickling: send to a subset once a period
- diffusion: send to all after random delays

Previous work

- Biryukov, Khovratovich, Pustogarov (2014) - "Deanonymisation of clients in Bitcoin P2P network" proposed a method for linking Bitcoin txs to IPs
- Key idea: nodes connect to 8 random "entry nodes", the "entry set" is a fingerprint

Outline

Understanding relationships between transactions

- Connect to many nodes
- Log timestamps of received tx announcements
- Intuition: we will hear of new txs from Alice or her entry nodes faster than from other nodes

Parallel connections

- Nodes maintain 8 outgoing and 117 (optional) incoming connections
- Txs propagate to some neighbors with random delays
- If we connect to a node once, the probability of getting a new tx quickly is low
- Can we connect to nodes many times in parallel?

Saturating connection slots

- `bcclient` tool connects to Bitcoin nodes with many parallel connections
- We occupy all available slots (avg 64 slots / peer on Bitcoin testnet)
- Nodes don't distinguish incoming and outgoing connections for tx propagation! Occupy 50% of slots – 50% chance of getting a new txs first.

Weighting timing vectors

- Earlier work only considered the *first* IP to relay a tx
- We consider the *vector* of the first 3 – 7 IPs to relay a tx, and assign them exponentially decreasing weights
- High correlation between vectors indicate the same originator

Weighting formula

IPs p_i get decreasing weights; median IP gets weight 0.5:

$$w(p_i) = e^{-(t_i/k)^2}$$

where

$$k = \frac{t_{median}}{\sqrt{-\ln(0.5)}}$$

Weighting timing vectors: example

High values indicate higher probability of an IP to be the originator or one of its entry nodes.

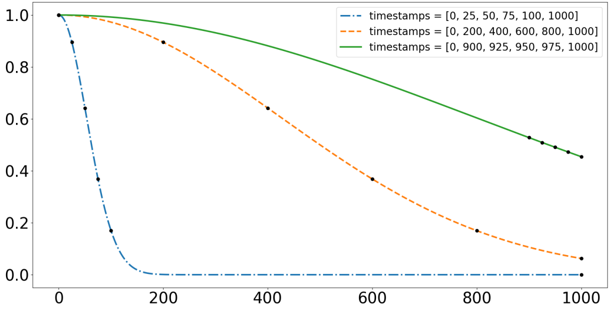


Figure: Weight function for 3 vectors of timestamps

Clustering of vectors

- For each pair of txs, calculate correlation of weight vectors
- Hypothesis: correlation matrix has a *block-diagonal* structure
- Related transactions form clusters along the main diagonal

Measuring clustering quality

Clustering algorithms decides for each pair of txs whether to put them in one cluster. Rand score reflects the share of right decisions:

$$R = \frac{SS + DD}{SS + SD + DS + DD}$$

where

- SS: same category, same cluster
- DD: different category, different cluster
- SD: same category, different cluster
- DS: different category, same cluster

Measuring anonymity

Anonymity degree measures the amount of information an attacker gains compared to *perfect anonymity*:

$$d = \frac{-\sum_{i=1}^{N} p_i log_2(p_i)}{log_2(N)}$$

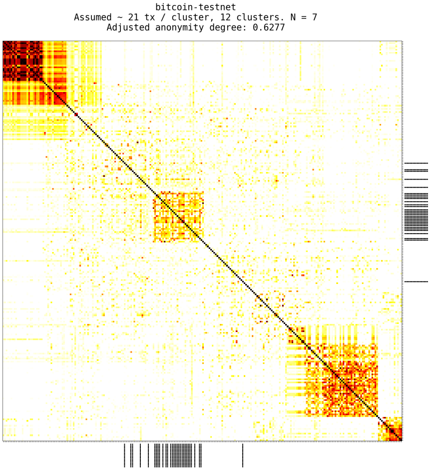
- $d = 1$: each user has an equal probability of being the originator of a given message
- $d = 0$: the attacker knows exactly the originators of all messages

Outline

Putting the pieces together

- Connect to many nodes in parallel, log tx announcements (use geographically distributed servers for better view of the network)
- Assign weights to vectors of timestamps
- Calculate correlations between pairs of weight vectors
- Apply a spectral clustering algorithm (`sklearn`)
- Choose best parameters from "learning set" of txs
- Calculate anonymity degree on "control set" of txs

Experiment (Bitcoin testnet)



Black lines: control txs. d: 0.63, precicion: 0.75, recall: 0.8.

Outline

Discussion

- Tx announcement timings reveal relationships between transactions, even with diffusion
- The technique works on testnet, worse on mainnet (though we didn't try to perform a full-scale attack)
- Cryptographic defenses (ZKPs, etc) don't work: we don't consider tx content

Countermeasures

- For users
  - Don't issue many txs in the same session
  - Run nodes with increased number of connection
- For cryptocurrency developers
  - Implement stronger broadcast randomization
  - Periodically drop and re-establish connections randomly
  - Increase the default number of connections

Of course, there are performance trade-offs.

New propagation mechanism for Bitcoin

- Dandelion: a proposal for new propagation mechanism for Bitcoin (BIP 156)
- Defeats our attack by distinguishing incoming and outgoing connections (it's hard to force a remote node to connect to us)

Outline

Alternative cryptocurrencies

- In this work, we only consider Bitcoin.
- Does our technique apply to coins other than Bitcoin? Some coins are based on Bitcoin's codebase (Zcash), some are not (Monero).
- How good is network-level privacy in other coins?

Mobile wallets

- In our experiments, txs were issues from a full node.
- Does the technique apply to transactions issued from mobile wallets?
- How are mobile wallets different in terms of networking?

Questions?

- cryptolux.org
- s-tikhomirov.github.io

uni.lu
UNIVERSITÉ DU LUXEMBOURG