Security and Privacy of Blockchain Protocols and Applications

Sergei Tikhomirov

UNIVERSITÉ DU LUXEMBOURG

Esch-sur-Alzette, Luxembourg, 17 September 2020

Part 1

Introduction

Problems with government-controlled money

- Unpredictable issuance
- Censorship and surveillance
- Political tool

“maintaining the dollar’s supremacy <...> is a critical strategic matter <...>. It is what allows us to have such effective sanction regimes around the world” – US Senator Tom Cotton (source)

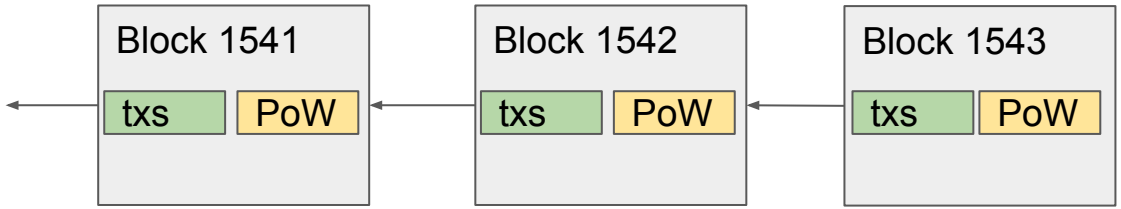
A long way towards decentralized digital money

- eCash (David Chaum, 1982)
- Hashcash (Adam Back, 1997)
- bMoney (Wei Dai, 1998)
- RPOW (Hal Finney, 2004)
- Bit Gold (Nick Szabo, 2005)
- Bitcoin (Satoshi Nakamoto, 2008)

“Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to **prevent double-spending**.”

Bitcoin in a nutshell

- Nodes broadcast transactions to a P2P network
- *Miners* produce *blocks* of transactions linked in a *chain*
- Each block *proves* that computational *work* has been performed to produce it
- Nodes choose the *heaviest* valid chain ⇒ consensus w/o trusted parties
- New coins enter circulation as miners' rewards on a predictable schedule



Challenges for cryptocurrencies

- Privacy. Transactions are broadcast and stored in plaintext.

Defenses against blockchain analysis (e.g., ZK) and **network analysis**.

- Scalability. All nodes validate all transactions.

On-chain tweaks, alternative blockchains, and **off-chain protocols** (Lightning).

- Programmability. Bitcoin’s Script is (intentionally) limited.

Alternative blockchains (Ethereum), smart contract programming languages.

Outline of this presentation

- Network-level privacy in Bitcoin and privacy-focused cryptocurrencies

A well-connected adversary can cluster transactions issued from the same node.

- Security and privacy of the Lightning Network

Privacy attacks on LN are likely if a few “important” nodes are compromised.

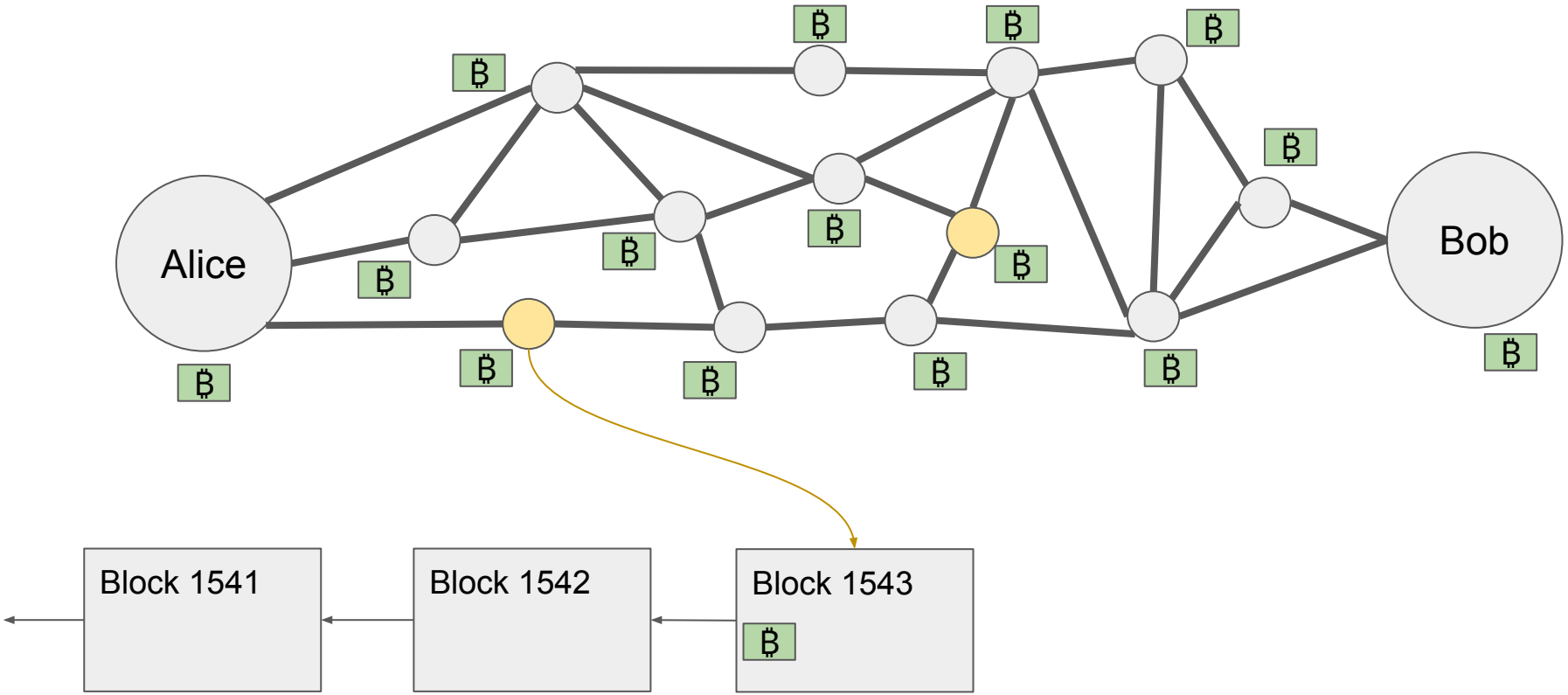
Lightning’s throughput is limited for small payments.

This limitation enables a new denial-of-service attack.

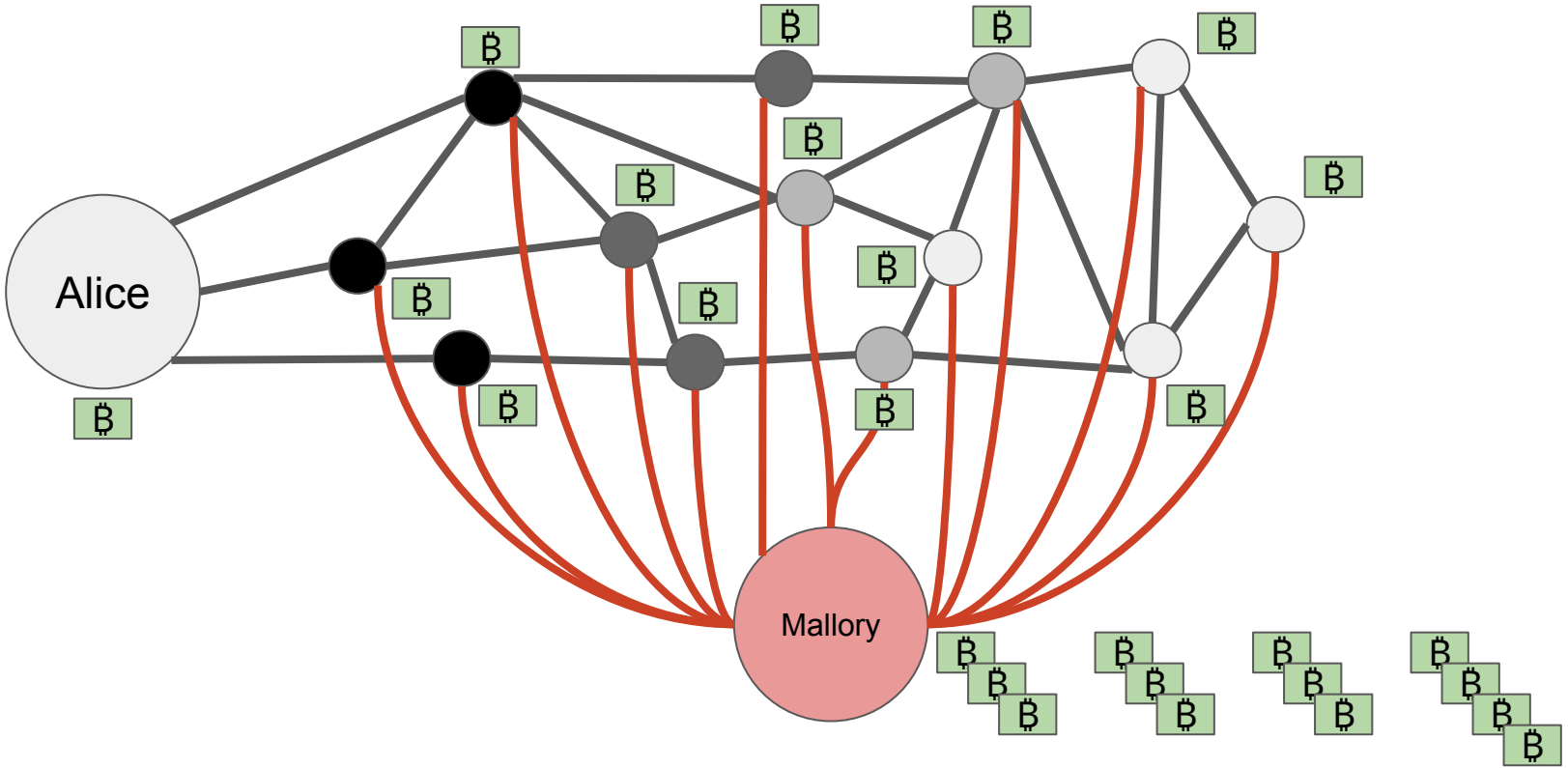
Part 2

Transaction clustering in Bitcoin and privacy-focused cryptocurrencies

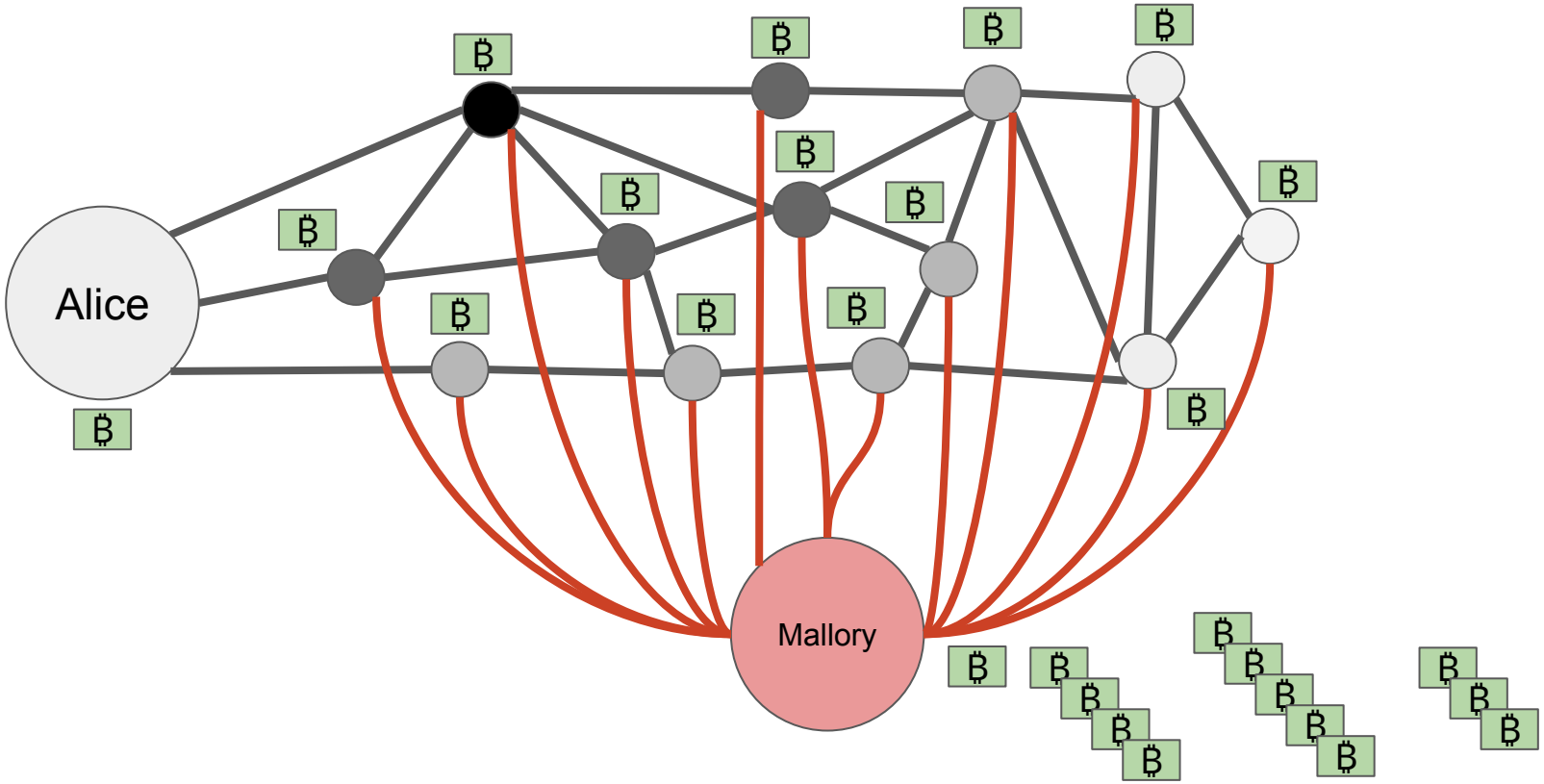
Transaction propagation



Transaction propagation



Broadcast randomization



Our contributions

We show how a network adversary can link transactions issued from one node.

Key idea: transactions from the same issuer exhibit similar propagation patterns.

The plan:

1. Define the “transaction propagation pattern”
2. Quantify the “degree of similarity” between propagation patterns
3. Cluster transactions based on their propagation patterns
4. Measure the resulting decrease in anonymity

One transaction from Mallory's perspective

IP address	**Received at (ms)**	**How likely to be "close" to the sender?**
IP_1	0	Highly likely
IP_2	10	Highly likely
IP_3	50	Likely
...	...	...
IP_{100}	5000	Highly unlikely

Transaction propagation vector

For each transaction, first IPs to announce it are likely to be “close” to the sender.

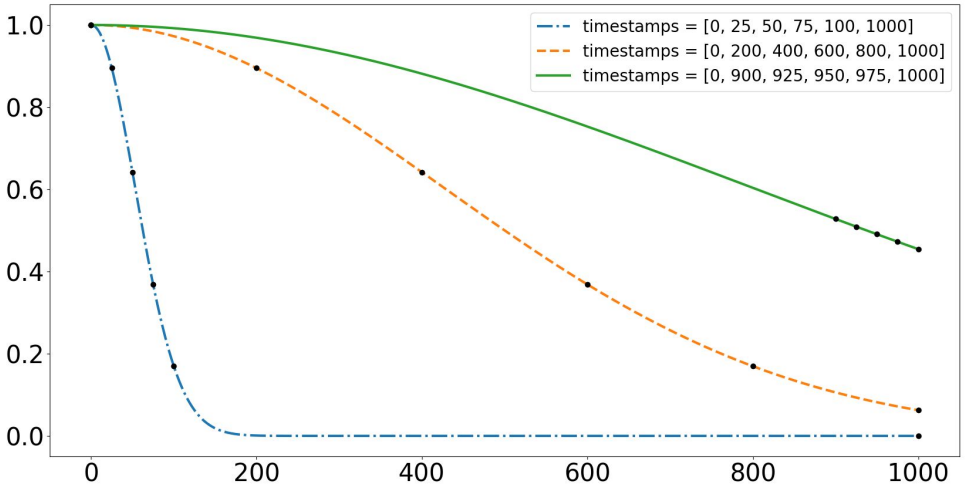
We assign weights to IP addresses based on timestamps of announcements:

- Weight = 1 for the first
- Weight = 0 for (N+1)-th and all the following

Tx	**IP_1**	**IP_2**	**IP_3**	**IP_4**	**IP_5**	**IP_6**	**IP_7**	**…**	**IP□**
Time	0	t_2	t_3	t_4	t_5	t_6	t_7	…	t□
Weight	1	?	?	?	?	0	0	…	0

Weight function

w(t) = e^(-t/k)², where k is chosen such that w = 0.5 for the median timestamp:

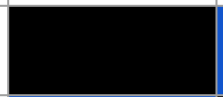
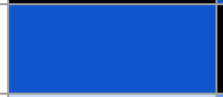
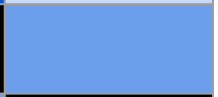
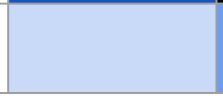
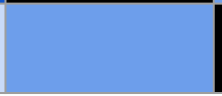
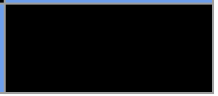


Intuition: timestamp difference is more important if the timestamps are near zero.

Comparing weight vectors

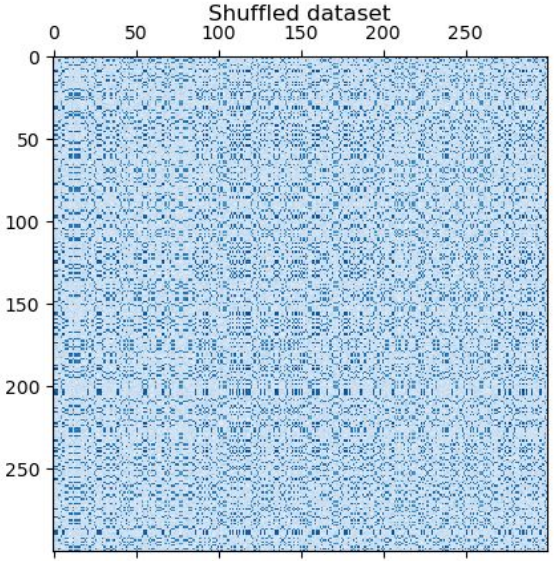
Tx	IP_1	IP_2	IP_3	IP_4	IP_5	IP_6	IP_7	IP_8	IP_9
0xa30e			1	0.3	0.5	0.1		0.7	
0x35a6	1		0.1	0.5	0.2	0.9			
0x196c		1					0.5		0.1

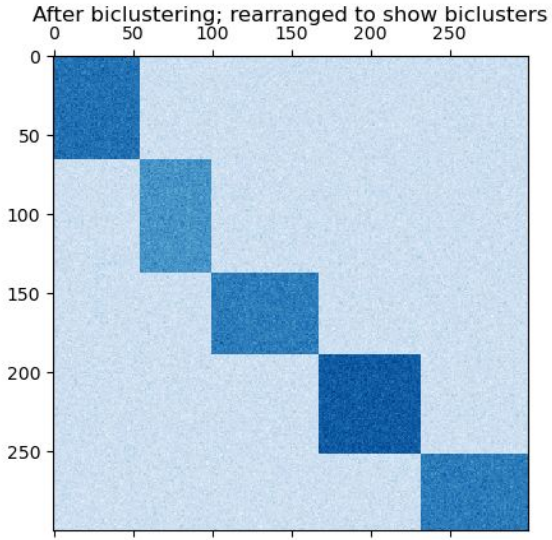
Tx	0xa30e	0x35a6	0x196c
0xa30e	1	-0.29	-0.45
0x35a6	-0.29	1	-0.43
0x196c	-0.45	-0.43	1

Tx	0xa30e	0x35a6	0x196c
0xa30e			
0x35a6			
0x196c			

Correlation matrix clustering

With a right row-column permutation, clusters become visible.





Source: https://scikit-learn.org/stable/auto_examples/bicluster/plot_spectral_coclustering.html

Ground truth with our own transactions

We use our own transactions as ground truth:

- Issue ~40 transactions from two nodes
- Divide our transactions into “learning” and “control” sets
- Run the clustering algorithm assuming the knowledge of the “learning” set
- Assess the result based on how well the “control” transactions are clustered

Measuring anonymity

We use anonymity degree proposed* by Díaz et al.:

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

Where p_i is the estimated probability of the i-th tx to originate from the control set.

- d = 1: full anonymity
- d = 0: no anonymity

* Díaz, Seys, Claessens, Preneel. Towards measuring anonymity. 2002

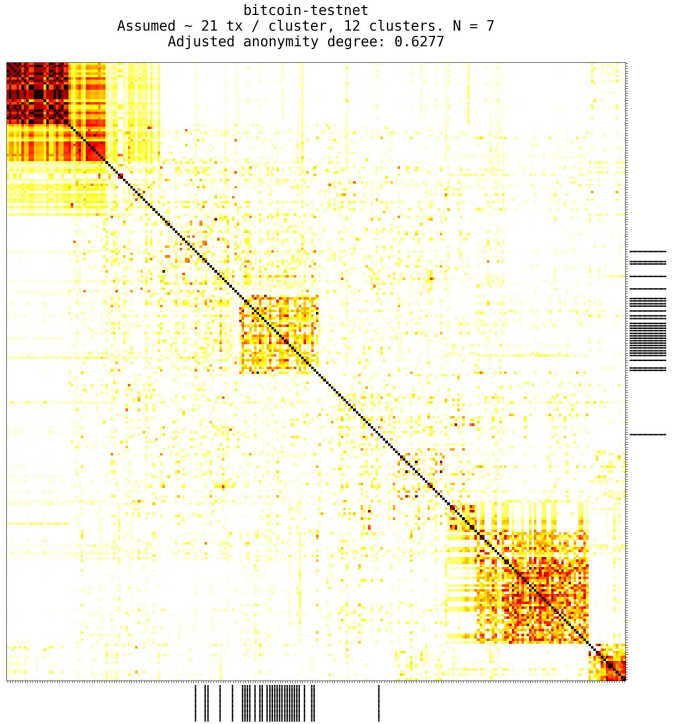
Experiment outline

Putting all the pieces together:

1. Launch three well-connected, geographically distributed listening nodes
2. Log all transaction announcements, including the learning and control sets
3. Assign weights to vectors of IP addresses for each announcement
4. Calculate pairwise correlations between the weight vectors
5. Apply the spectral co-clustering algorithm to the correlation matrix
6. Calculate the anonymity degree using our own transactions

Let’s look at the results...

Bitcoin testnet. Anonymity degree = 0.63



Privacy-focused cryptocurrencies

Dash: Bitcoin Core fork

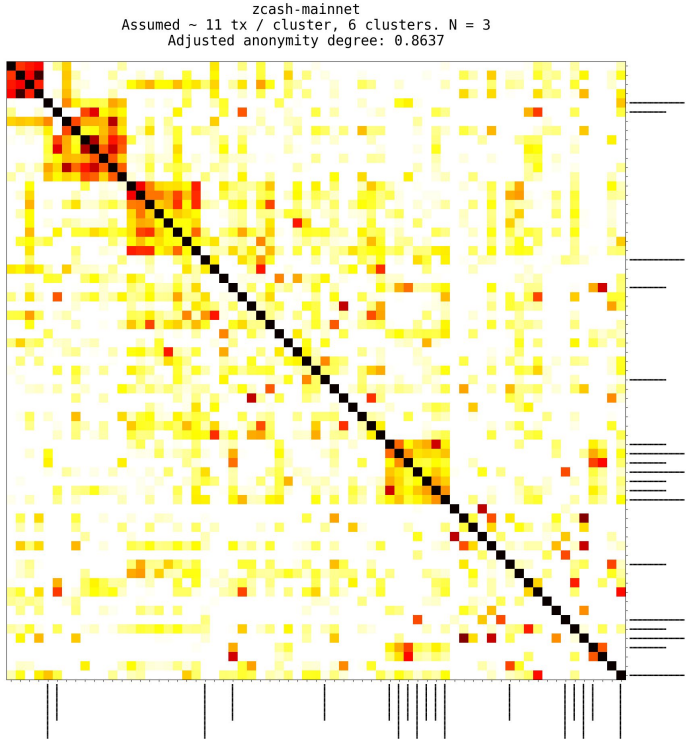
Monero: implemented from scratch

Zcash: Bitcoin Core fork

Various cryptographic and application-level techniques are used.

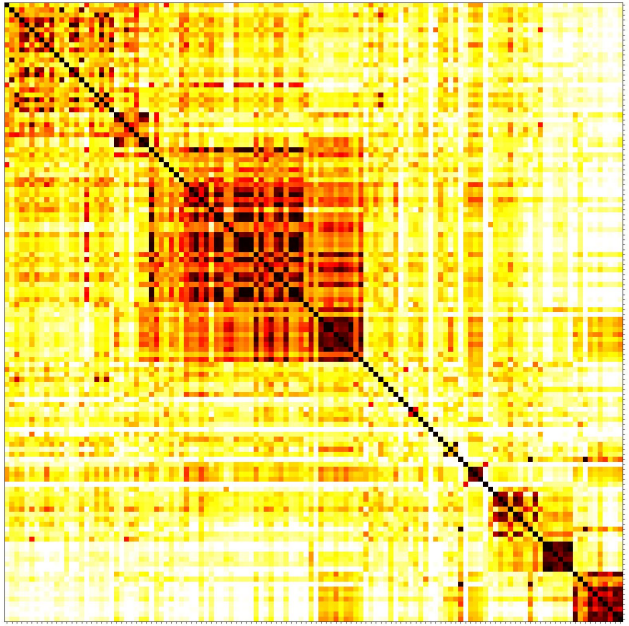
What about network-level privacy?

Zcash. Anonymity degree = 0.86

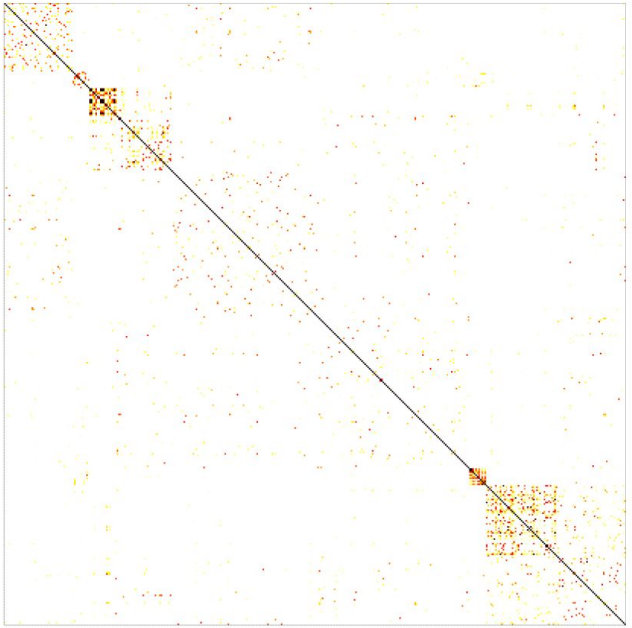


Monero and Dash

monero-mainnet
Assumed ~ 12 tx / cluster, 10 clusters. N = 3



dash-mainnet. N = 4, 9 clusters



Summary of part 2

P2P traffic reveals links between transactions from one sender.

Advice for users:

- Don’t issue multiple transactions from the same session
- Run nodes with an increased number of connections
- Periodically drop random connections and establish new ones

Advice for developers:

- Implement stronger broadcast randomization
- New P2P protocols: Dandelion and Erlay
    - Both prevent our attack: transactions are initially sent to outgoing connections only

Part 3

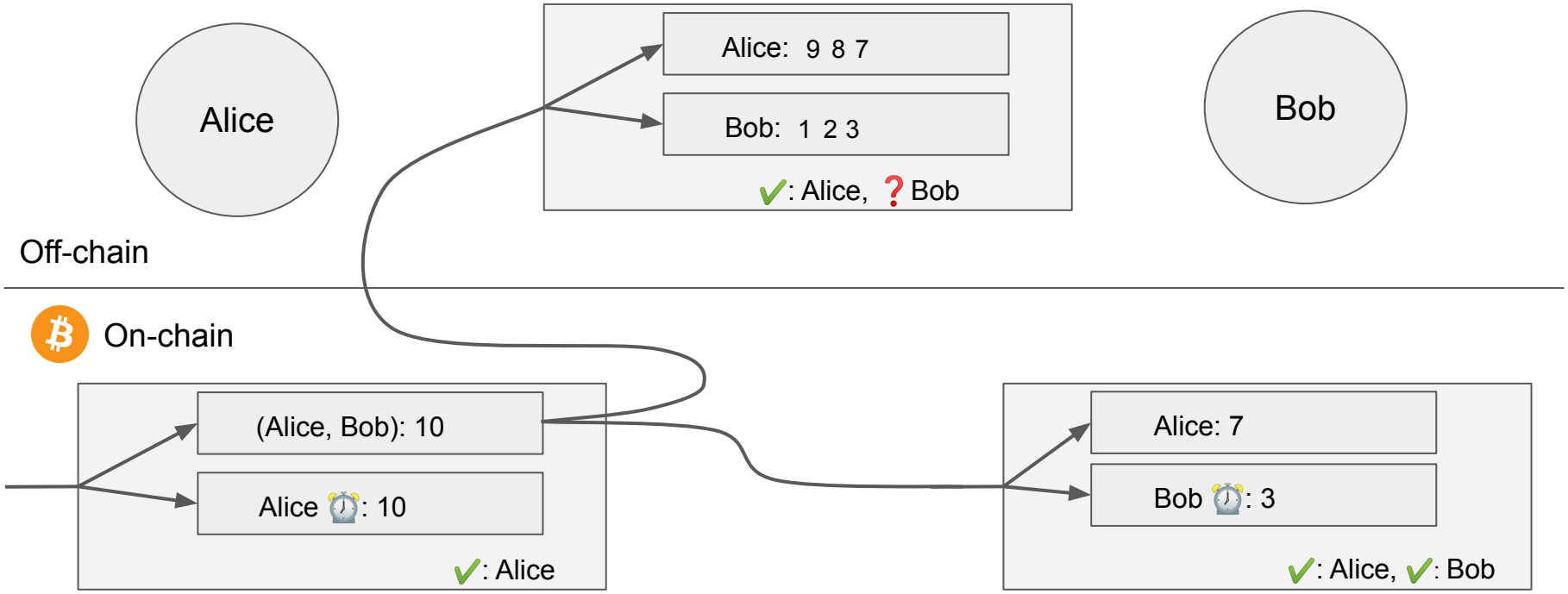
Security and privacy of the Lightning Network

Joint work with Pedro Moreno-Sanchez and Matteo Maffei (TU Wien)

Off-chain protocols

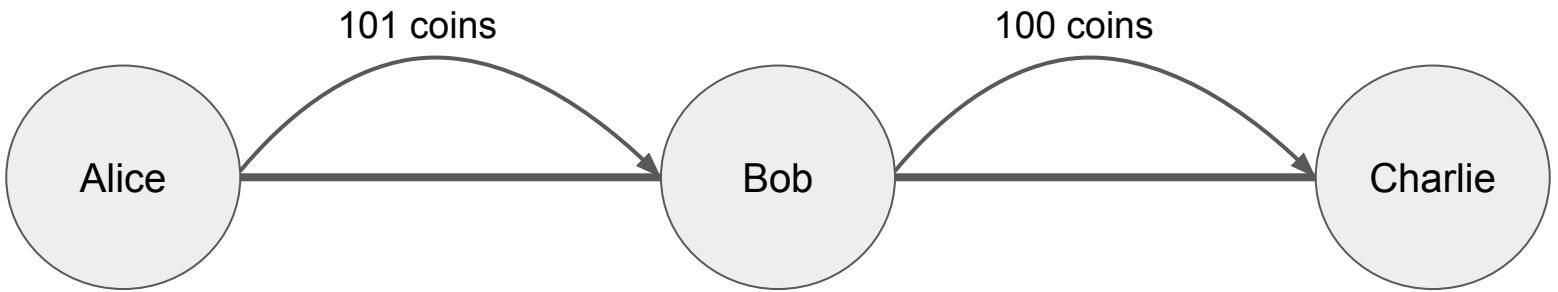
- Idea: let’s move (most of the) transactions *off-chain*
- Pros: high throughput, no changes to the main protocol
- Cons: new security and privacy challenges

Payment channel example



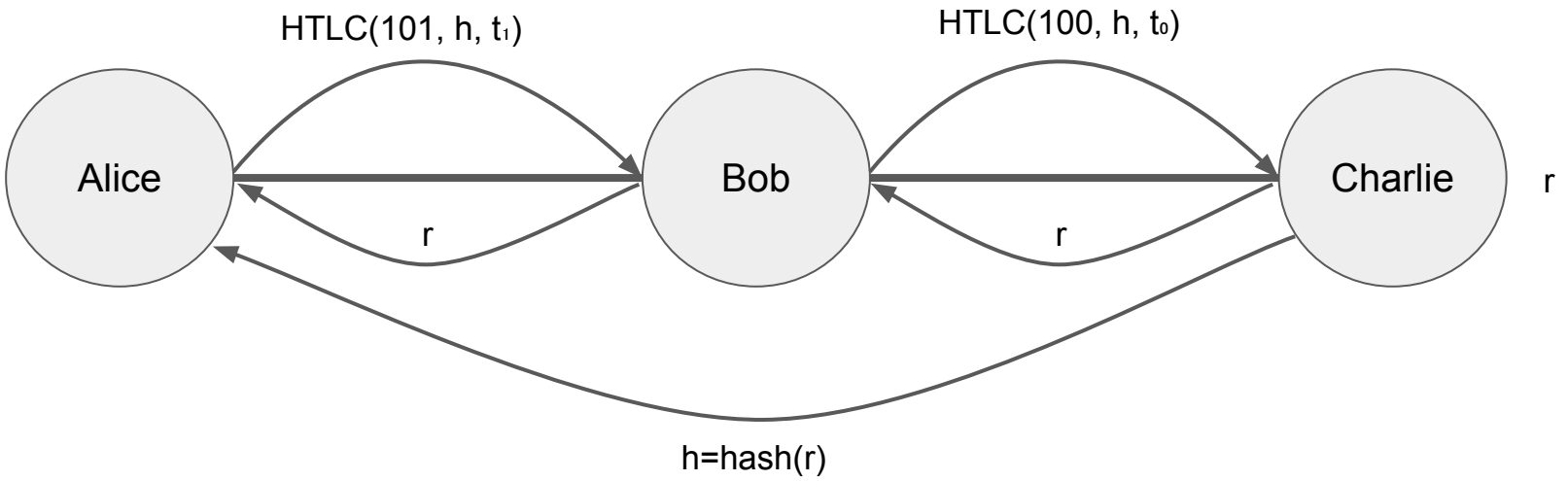
Payment channel network

- Expensive to open channels between every two users (fees, confirmations)
- Solution: a network of payment channels
- Must ensure atomicity in multi-hop payments



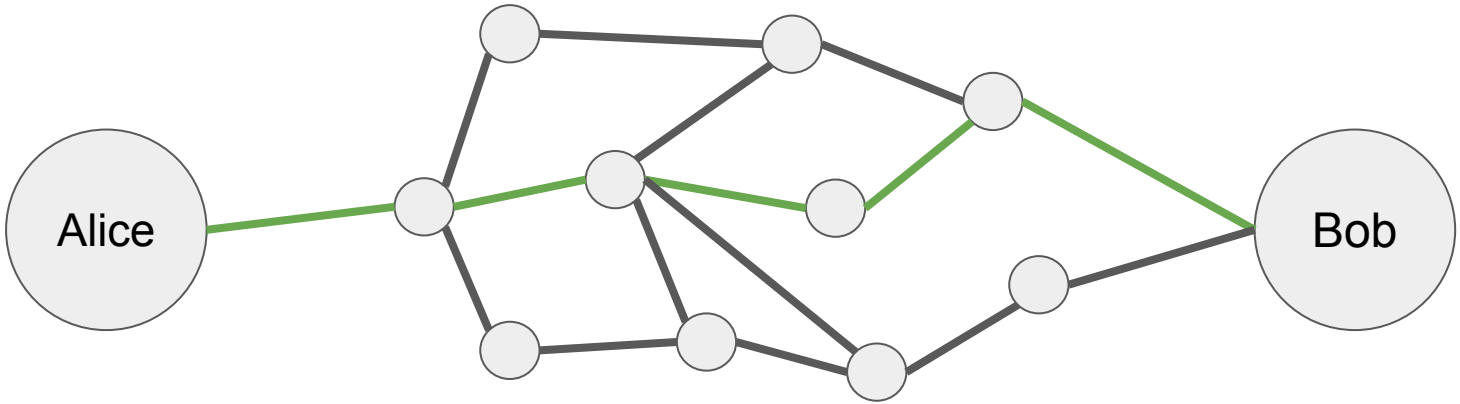
Lightning Network architecture

- LN ensures atomicity with hash time-locked contracts (HTLCs)
- Coins go to Bob if he shows a hash preimage before time t, otherwise to Alice



Source routing

- Nodes gossip about channels available for routing
- Each node compiles a local view of the network
- The sender chooses the route based on the local view
- If a payment fails, the sender tries another route



Our contributions

What do LN's security, privacy, and throughput depend upon?

In this work, we:

- quantify the effect of three previously described attacks*
- analyze a limitation on payment concurrency
- describe a new DoS attack vector

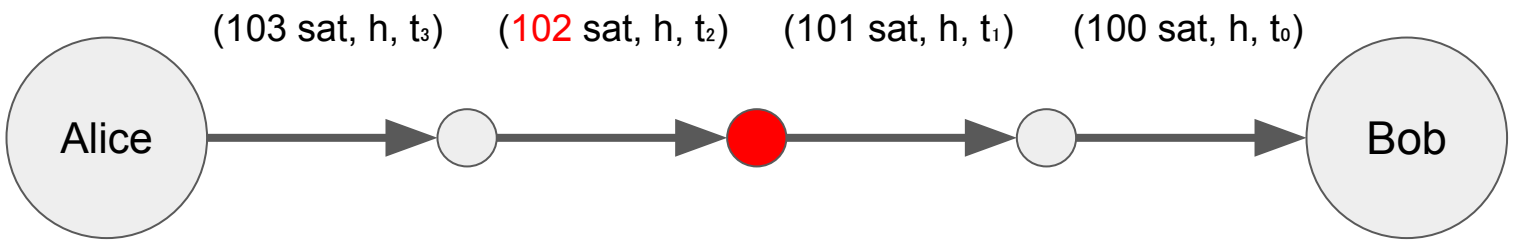
* Malavolta et al. Concurrency and privacy with payment-channel networks. CCS, 2017.
Malavolta et al. Anonymous multi-hop locks for blockchain scalability and interoperability. NDSS, 2019.

Value privacy

Attacker learns how much is being transacted.

Trivial for on-path adversaries: amounts are in plaintext.

Sufficient condition: 1 attacker's node on the path.

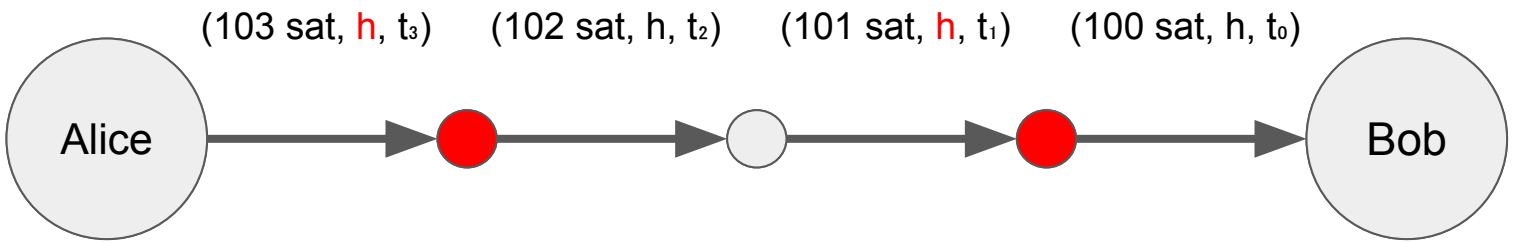


Relationship anonymity

Attacker learns who pays whom (with probability much better than random guess)

Payments are linked by the same hash value.

Sufficient condition: 2 attacker's nodes on the path: the first and the last.

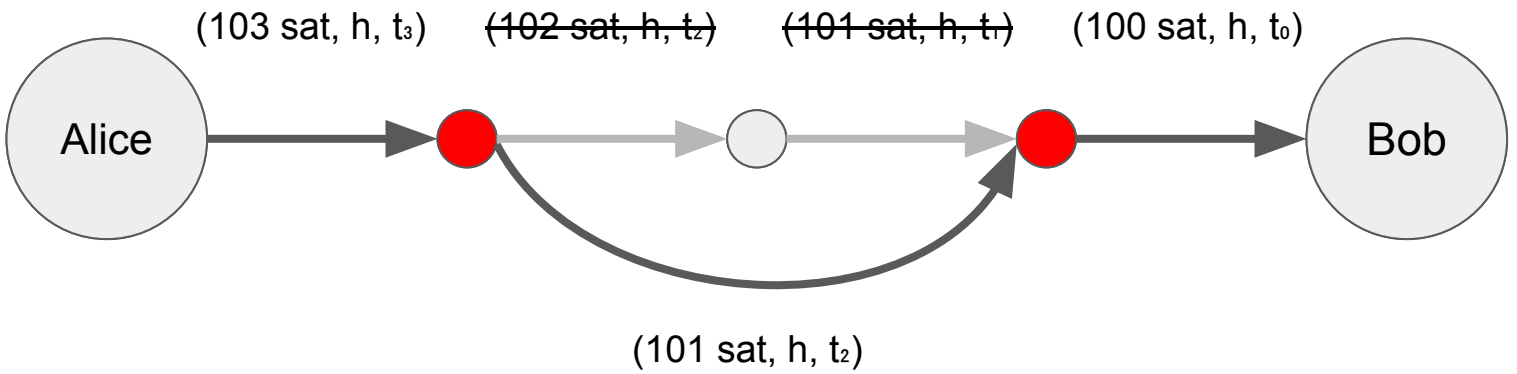


Wormhole attack

Attacker “shortcuts” a payment, taking fee from the honest node.

Damage for the honest node: a) no fees, b) capital locked until timeout expires.

Sufficient condition: 2 attacker’s nodes on the path with honest nodes in between.



How likely is an attack to succeed?

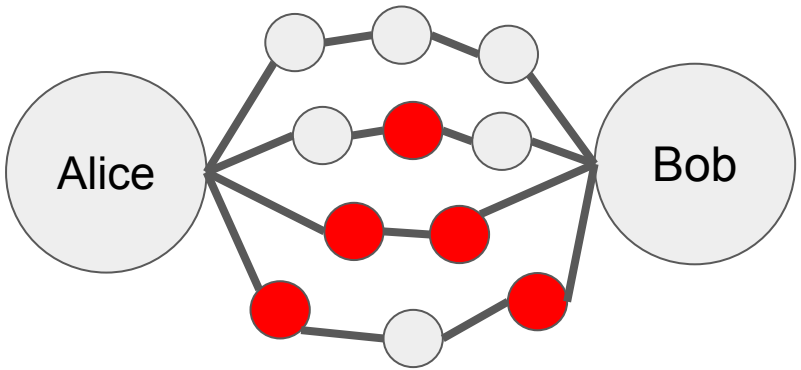
It depends on various factors:

- The type of the attack
- The payment amount
  - Smaller payments have more routing options
- How many nodes are compromised
- *Which* nodes are compromised

We aim to quantify the importance of these factors based on real LN data.

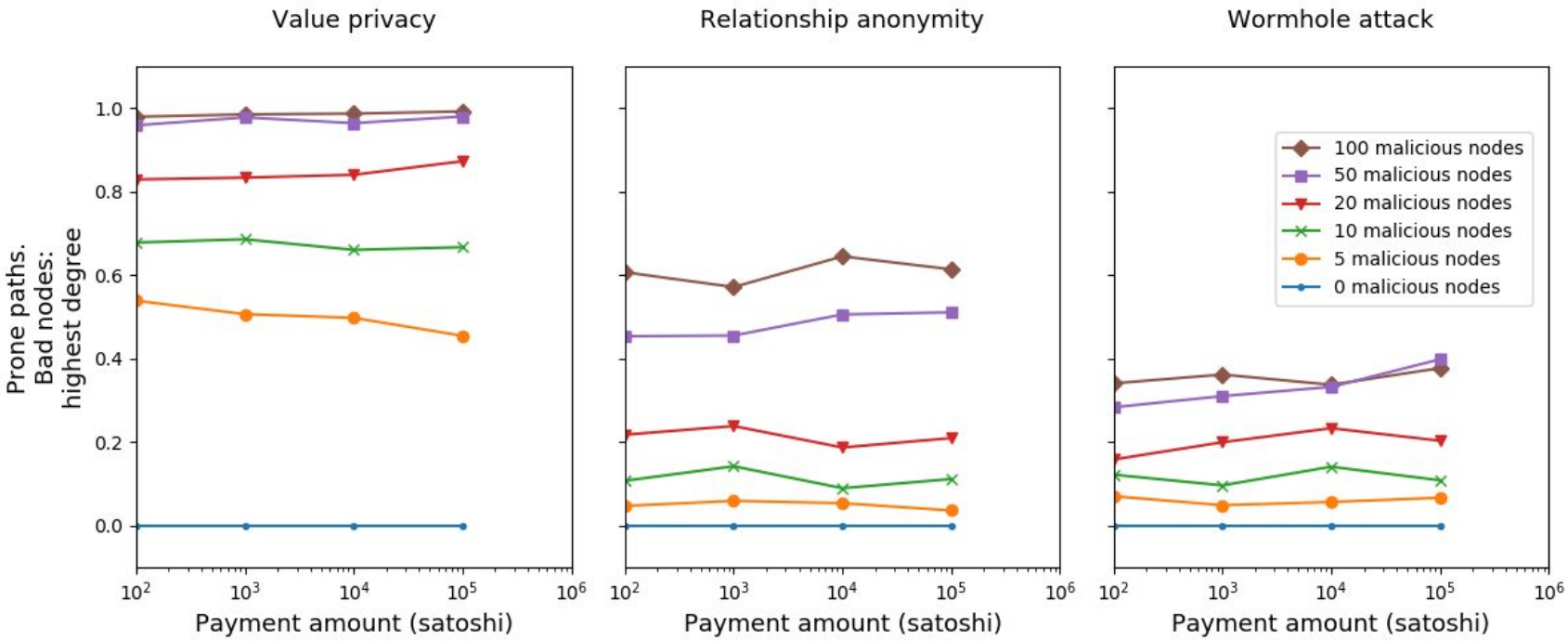
Experiment outline

- Assume that a certain subset of nodes is compromised
- Find all suitable paths between random sender and receiver
- Calculate the share of paths vulnerable to a given attack
- Average the result across many random runs

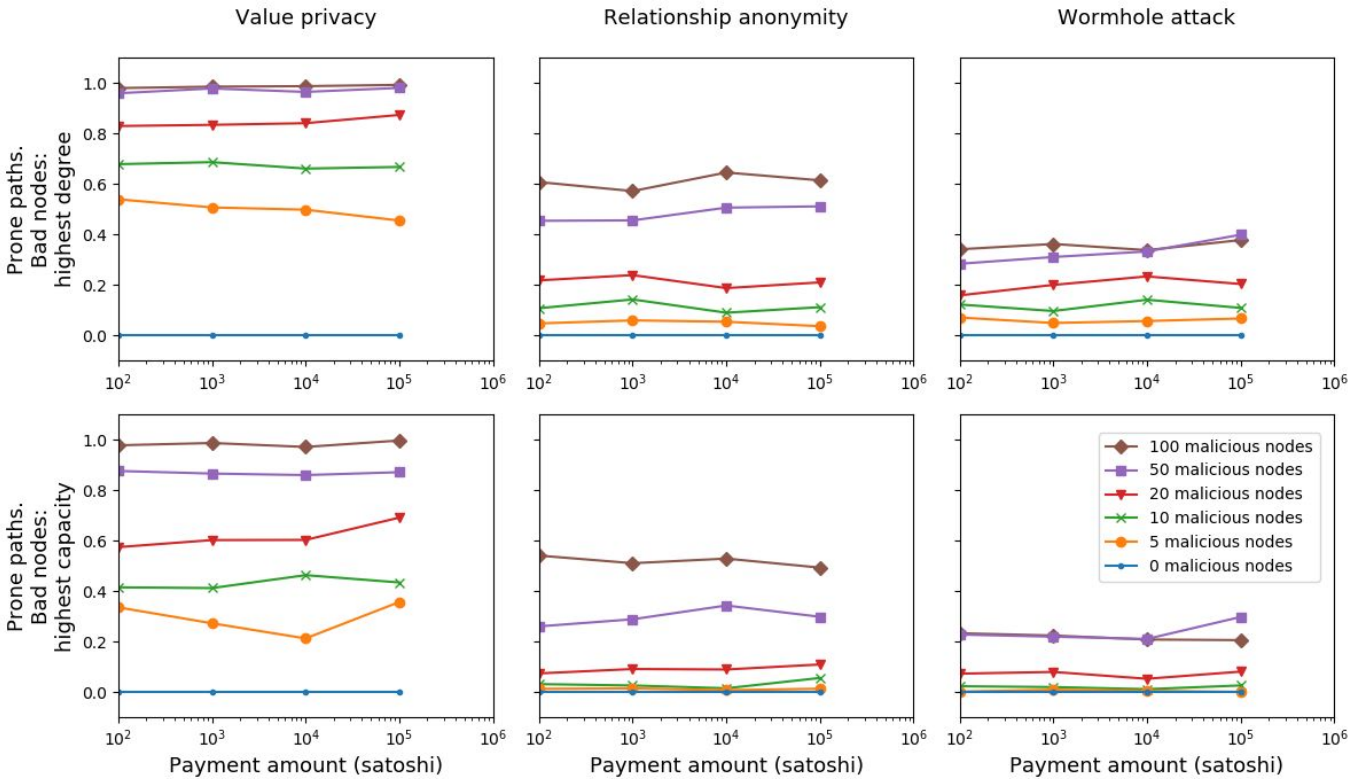


	VP	RA	WA
Path 1	Safe	Safe	Safe
Path 2	Prone	Safe	Safe
Path 3	Prone	Prone	Safe
Path 4	Prone	Prone	Prone
Prone	75%	50%	25%

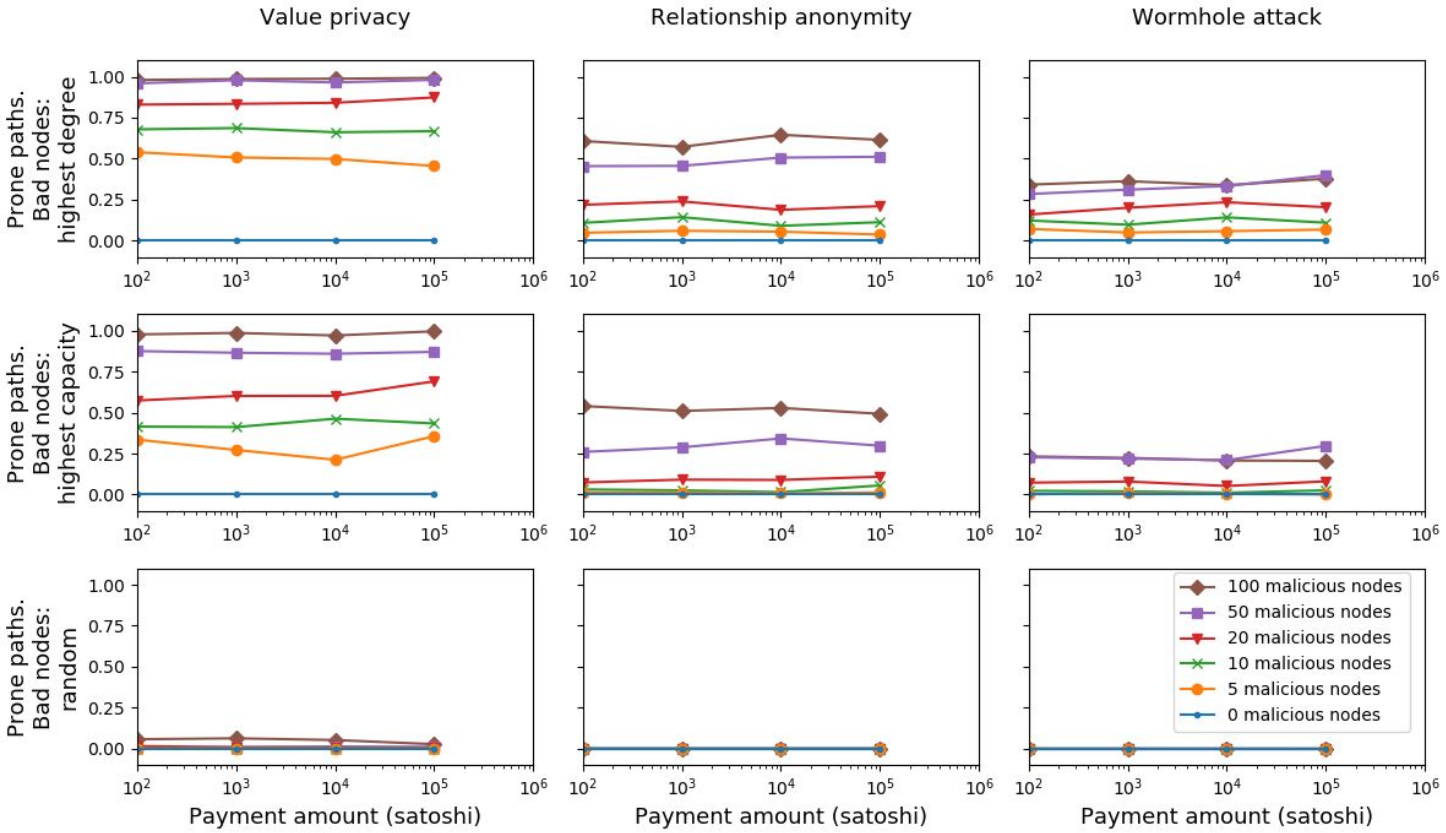
Results: highest degree nodes compromised



Results: + highest capacity nodes compromised

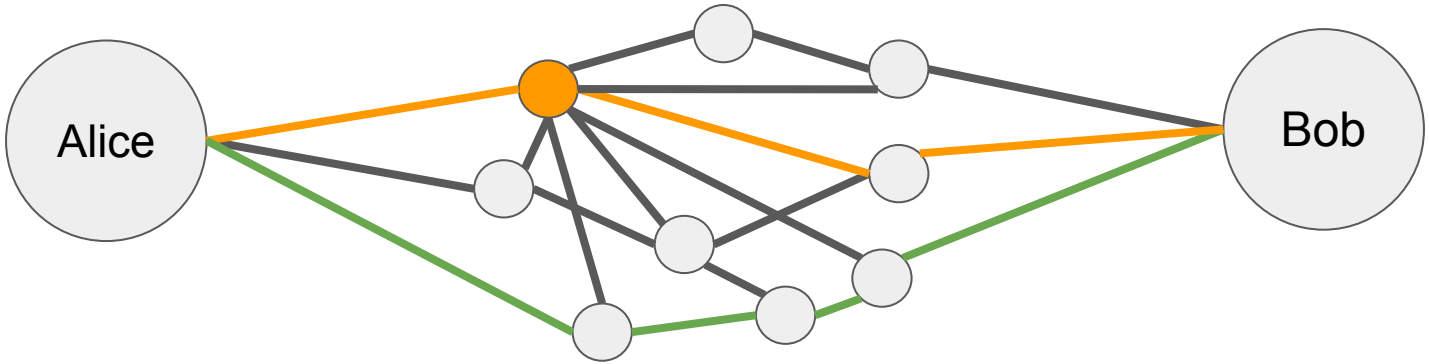


Results: + random nodes compromised



Trade-off between connectivity and privacy

- Routing via large nodes: dangerous if they are compromised
- Routing via small nodes: less liquidity and uptime



HTLC limit

How many concurrent payments can LN handle?

- One channel may hold multiple unresolved HTLCs
- Channel parties must be able to dispute malicious closures on-chain
- Dispute transactions include all unresolved HTLCs
- Bitcoin transactions must be < 100 KB
- Consequently, a channel supports at most 966 HTLCs (*HTLC limit*)

Example of HTLC depletion

Consider a channel with capacity of 1M sat.

967-th HTLC cannot be added, though the capacity is not depleted.

	Unresolved HTLCs
1	HTLC (to Alice, 1000 sat, 0xdf86...)
2	HTLC (to Bob, 1000 sat, 0x0a1f...)
…	…
966	HTLC (to Alice, 1000 sat, 0x6f26...)
Total value of HTLCs (sat)	966k < 1M
Number of HTLCs	966

Limiting factors for throughput

Two limiting factors: channel capacity and HTLC limit.

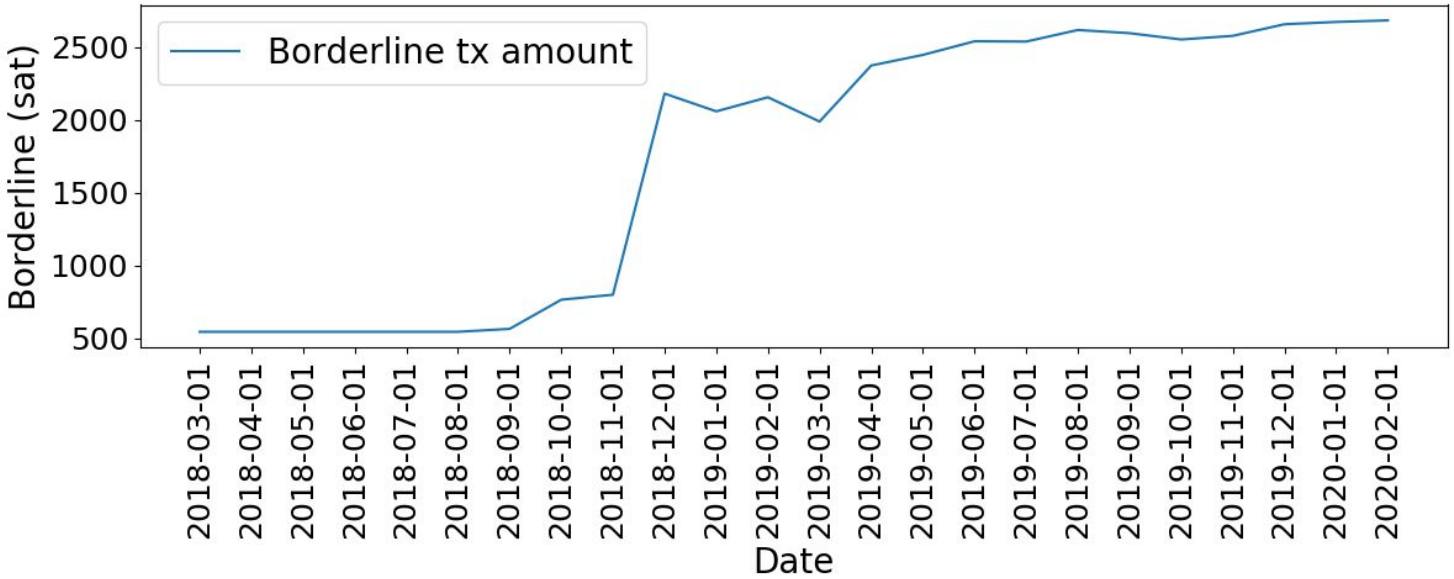
Which one is more important? It depends on the amount:

- 0 – 546 sat (dust limit): no HTLC created
- 546 – **2700 sat** (0.3 USD): HTLC limit is more important
- >2700 sat: capacity is more important

We refer to 2700 sat as the **borderline amount**.

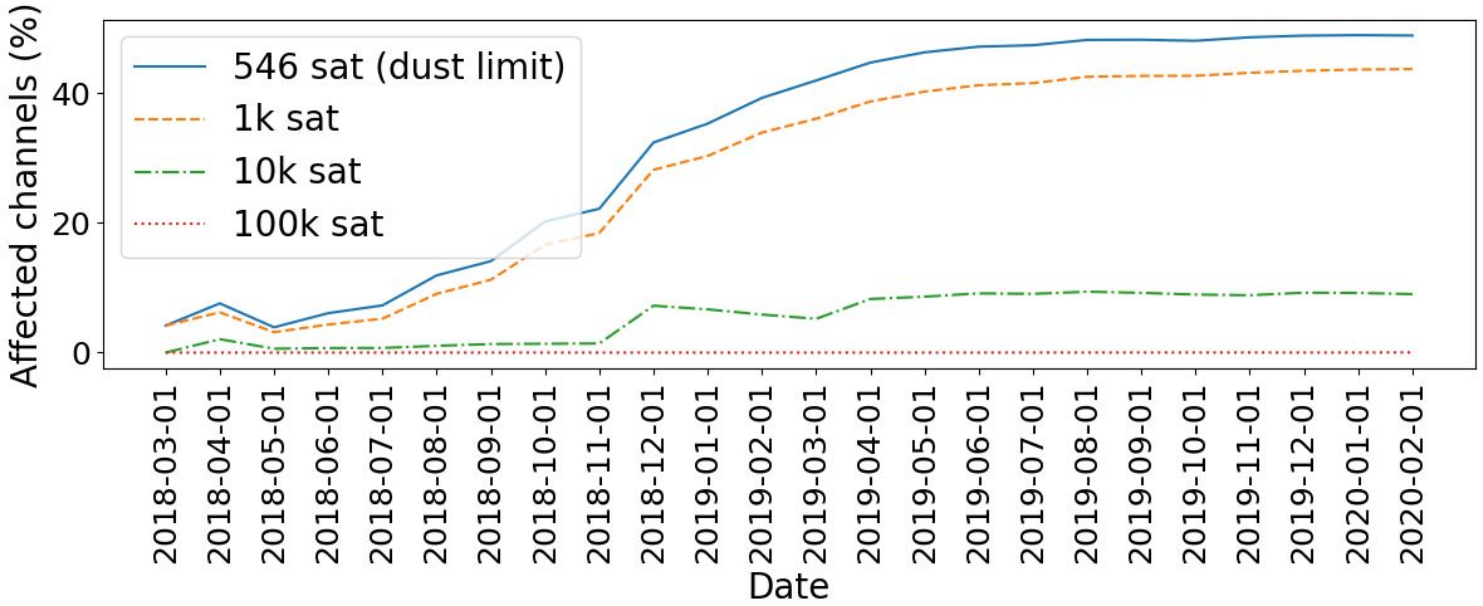
Evolution of borderline amount

Borderline amount relatively stable since 2019.



Up to 50% of channels affected

Nearly half of all channels could have handled more concurrent micropayments.



DoS by exceeding the HTLC limit

- An attacker blocks a channel by sending 966 near-dust payments
- Does not require as many coins as in the victim channel
- Can block one channel with 966*546 = 527k sat (~60 USD)
  - Can block N times more more with an N-hop payment

Channel capacity (sat)	**Cost of depleting one channel (sat)**	
	Capacity-based	**HTLC-based**
100k	100k	527k
1M	1M	527k
10M	10M	527k

Summary of part 3

- Privacy attacks are possible with only a few “important” nodes compromised
  - In September 2019, one entity (LNBIG) controlled 40% of LN capacity
- Throughput is constrained by the HTLC limit, in addition to channel capacity
- HTLC limit is relevant for micropayments (< 0.3 USD)
- A new DoS vector allows to block large channels cheaply

List of publications

1. Biryukov, Khovratovich, Tikhomirov. “Findel: Secure Derivative Contracts for Ethereum”. WTCS@FC 2017
2. Tikhomirov. “Ethereum: State of Knowledge and Research Perspectives”. FPS 2017
3. Biryukov, Khovratovich, Tikhomirov. “Privacy-preserving KYC on Ethereum”. ERCIM-Blockchain 2018
4. Tikhomirov, Voskresenskaya, Ivanitskiy, Takhaviev, Marchenko, Aleksandrov. “SmartCheck: Static Analysis of Ethereum Smart Contracts”. WETSEB@ICSE 2018
5. Biryukov, Tikhomirov. “Transaction Clustering Using Network Traffic Analysis for Bitcoin and Derived Blockchains”. CryBlock@INFOCOMM 2019
6. **Biryukov, Tikhomirov. “Deanonymization and Linkability of Cryptocurrency Transactions Based on Network Analysis”. EuroS&P 2019**
7. Biryukov, Tikhomirov. “Security and privacy of mobile wallet users in Bitcoin, Dash, Monero, and Zcash”. PMC #59, 2019
8. **Tikhomirov, Moreno-Sanchez, Maffei. “A Quantitative Analysis of Security, Anonymity and Scalability for the Lightning Network”. S&B@EuroS&P 2020**
9. Tikhomirov, Pickhardt, Biryukov, Nowostawski. “Probing Channel Balances in the Lightning Network”. 2020

Thank you! Questions?

Feel free to reach out:

- Twitter: @serg_tikhomirov
- Telegram: @s_tikhomirov
- sergey.s.tikhomirov@gmail.com

Image sources

- https://explorer.acinq.co/
- https://twitter.com/JustinAHorwitz/status/1248384733156741123/photo/1
- https://www.reddit.com//fnchzr/
- https://en.bitcoin.it/wiki/Promotional_graphics
- https://twitter.com/ValueOfBitcoin/status/1258887539903148032