

On privacy metrics for multi-hop payments in the Lightning Network

G. Navarro-Arribas¹[0000-1111-2222-3333] and
Kanishkar Kannan²[0009-0003-4392-9784]

¹ Universitat Autònoma de Barcelona

`guillermo.navarro@uab.cat`

² SASTRA Deemed University

`kanishkark1411@gmail.com`

Abstract. The Lightning Network, is a payment network that can operate on top of Bitcoin. It provides the ability to perform multi-hop payments, with some privacy guarantees. These include the fact that an intermediate node in a multi-hop payment cannot know the actual source and destination of the payment. In this paper, we propose metrics to quantify the effective privacy achieved in such scenarios, explicitly accounting for adversarial inference enhanced by heuristic techniques. We conduct an empirical evaluation of these metrics to assess their accuracy and practical relevance.

Keywords: Payment network · Privacy · Routing.

1 Introduction

The Lightning Network (LN) [14] was built to overcome the scalability and latency problems of Bitcoin (and similar cryptocurrencies). The basic construct of the LN is the payment channel. A channel can be opened between any 2 parties by depositing some amount. The amount invested by a party is its balance, and the sum of the two balances is the capacity of the channel. Channel creation requires both parties to send a transaction (funding transaction) to the blockchain, but once established, they can exchange payments without relying them to the blockchain. This makes it possible to have fast payments at a relatively low cost. Each payment will update the balance of both parties, which will be committed to the blockchain when the channel is closed. Closing a channel can be done gracefully by both parties, or abruptly by one party in case of conflict, by a transaction on the blockchain (commitment transaction).

A key concept of the LN is that two nodes can perform a payment between them without having a direct channel. This is done by routing such payment through other channels of the network as a multi-hop payment. The intermediate nodes will collect some fees as an incentive for their participation. The LN can thus be seen as a network of payment channels, and information about payment channels such as their capacity and availability is broadcasted to all the nodes

by using a gossip protocol. It is important to note that not all the information is made public. For instance, the balance of each channel is considered private in an attempt to hinder payment tracking. Nodes broadcast only the information needed by other nodes to find a payment path. This information includes the channel capacity and the fees, among other things.

In order to prevent intermediate nodes in a multi-hop payment to gain knowledge of the payment they are routing, the LN makes use of an onion routing approach to route the payment. An intermediate node only knows its predecessor and successor in the payment, but not the source and destination nodes. A middle node can however use publicly available information to attempt to infer the source and destination of the payment. Most notably, the public topology of the payment network.

This means that it might be possible to deanonymize the source and destination of a multi-hop payments, assuming the attacker is a node in the payment path. This is a known problem of payment networks such as LN. In this paper we elaborate on the use of the network topology and the fact that the algorithm used to determine the payment route is deterministic, in order to bound the possible origin and destination nodes of a multi-hop payment observed by an intermediate adversary.

We provide formal definitions, and show results using a snapshot of the LN network on Bitcoin. We also introduce the idea of using heuristic estimations based on observable attributes of the payment such as the expected amount of the payment.

The paper is organized as follows. Section 2 describes multi-hop payments in the Lightning Network and introduces the notation used in the paper and some related work. Section 3 describes the adversarial model, and Section 4 provides a description and definitions of the privacy metrics. Section 5 presents our evaluation, and Section 6 concludes the paper.

2 Multi-hop payments in the Lightning Network

In this section we introduce briefly multi-hop payments in LN. We introduce a general pathfinding technique, which aims to provide a general baseline. The most popular implementations of the LN are LND³, Core Lightning⁴, Eclair⁵, and LDK⁶. Each implementation provides a slightly different pathfinding algorithm, but they present enough similarities. We present here an abstraction, providing a general framework for evaluation.

2.1 Multi-hop payments and HTLCs

Multi-hop payments in LN are possible thanks to the *Hashed Time-Locked Contracts* (HTLC). An HTLC is a smart contract where a sender A locks funds that

³ <https://github.com/lightningnetwork/lnd>

⁴ <https://github.com/elementsproject/lightning>

⁵ <https://github.com/aciinq/eclair>

⁶ <https://lightningdevkit.org/>

a receiver B can claim by providing a digital signature and the secret preimage corresponding to a specific hash value. If B doesn't claim the funds and reveals the preimage before an expiration time, the *timelock*, A can reclaim the deposit with just its signature. For example, for a payment $A \rightarrow B \rightarrow C$, the process begins with the final recipient (C) generating a secret value (x) and sending its hash ($h(x)$) to the original sender (A). A then initiates a one-hop payment to B using an HTLC conditioned on $h(x)$. Simultaneously, B sets up a one-hop payment to C , also using an HTLC based on the same $h(x)$. Because C knows x , they can claim the payment from B . This action of claiming inherently reveals x to B , allowing B to then claim its payment from A .

To mitigate the risk of funds being locked indefinitely, the sender (A) specifies a total timelock for the entire payment. This time lock is set as an absolute block height on the blockchain, after which A can unilaterally recover its funds. As the payment is relayed through each intermediate node, each node reduces this time lock by a predetermined value, its *timelock delta*, which is publicly advertised for its channels. The payer calculates the initial timelock to ensure that even after subtracting the cumulative timelock deltas values of all intermediate nodes, the final node still has sufficient time to claim the payment before its HTLC expires.

A privacy measure used in multi-hop payments is the use of an onion routing scheme with Sphinx packet structure [2]. A node taking part in a multi-hop payment will only know the preceding and subsequent node in the payment path, all other nodes (possibly including source and destination) will be unknown.

2.2 Notation and payment path-finding algorithms

The LN is commonly modeled as a directed multigraph $G = (V, E)$, where each node is a vertex $v \in V$, and each payment channel is modeled as two directed symmetric edges. Given a payment channel between nodes u, v , we will have two edges $e_1 = (u, v)$, and $e_2 = (v, u)$, so $e_1, e_2 \in E$. Each edge can have different attribute values, since the public information of the channel can have different values for each direction on the same channel (e.g. fees, lock times, ...).

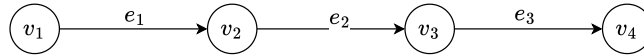


Fig. 1. Multi-hop payment example.

We will denote a payment path as a sequence of edges. For example, the path in Figure 1 will be denoted as $P = (e_1, e_2, e_3)$, or more generally $P = (e_1, \dots, e_k)$ as a path of $k + 1$ nodes and k edges. We use edges to denote the path since they are the building block of multi-hop payment in the Lightning Network. Moreover, we can have more than one channel between the same two nodes. We will denote the size of the path as $|P_{v_i v_j}|$ as the number of edges in the path.

When the node v_1 in Figure 1 wants to make a payment of a given amount x to v_4 , the initial amount provided by v_1 needs to account for the fees of

each intermediate node. The path is constructed from the source (v_1) and the amount for each edge (each corresponding HTLC) is computed backward (from destination) in order to determine the initial amount to be sent by v_1 .

The main goal when finding a payment path is to minimize the cost of the payment. Cost is commonly understood in a broad sense, and path-finding algorithms take into account the fees but also the cost that might be derived by a failed payment attempt, or the delay that might be produced due to channels being locked.

The specific details of the path-finding algorithm vary depending on the actual implementation, but they provide a similar approach. They use a variation of the Dijkstra shortest path algorithm, where the weight or cost for an edge is determined as a combination of the different attributes of the edges.

As a general approach, we consider that each edge, has an associate weight $w(e)$, and the cost of the path P will be:

$$C(P) = \sum_{e_i \in P} w(e_i)$$

We can generalize the weight of an edge $e \in E$, as:

$$w(e) = fees(e) + tldp(e) + bias(e)$$

Where, each field is:

- $fees(e)$ (fees): the fees collected by a node are determined by the *base fee* (fb_e), and the *fee rate* or *proportional fee* (fp_e) of the edge e for which the payment is received. The base fee is directly added, while the proportional fee is based on the actual amount being transferred. In general, $fees(e_i) = (fb_{e_i} + fp_{e_i} \cdot x_{e_i})$, where x_{e_i} is the amount received by the source node of edge e_i .
- $tldp(e)$ (timelock delta penalty): the weight function penalizes channels with a high timelock delta, δ_e , which is set by the receiving node of the channel. Most implementations use something like $tldp(e) = \delta_e \cdot x_e \cdot rf$, where rf is a risk factor configured by the administrator (e.g., LND uses a default value of $rf = 1.5 \times 10^{-8}$).
- $bias(e)$ (success probability): implementations use some estimation of success probability for using each edge of the path, which we generalize as a $bias_e$ for the edge e . This can take into account different factors depending on the concrete implementation:
 - Probability of success based on estimation of balance: LND uses 2 strategies. The apriori estimation determines the relationship between the intended payment amount and channel capacity, while the bimodal distribution highlights the idea that the balance is not evenly distributed, but mostly concentrated in one side of the channel [13,9,1]. Eclair uses a similar apriori approach, and CLN and LDK also do it but introducing an estimation of a lower and upper bound for the balance of the channel.
 - Virtual cost (*attempt cost* in LND or *failure cost* in Eclair): implementations estimate a virtual cost for the channel proportional to the previous probability and the amount being transferred.

- Channel age: some implementation such as Eclair also use the channel age (block height of the funding transaction), assuming that older channels are more reliable.

It is important to note that some implementations also use path related success probability factors. Since they are not associated to edges, they can not be used in the weight and cost function of the path-finding algorithm. This requires a slight modification of the Dijkstra algorithm. Other, as LND perform the Dijkstra backwards to precisely compute fees in each step. For a more exhaustive and detailed review on path-finding strategies for the LN, we refer the reader to [16,8] or the concrete implementations for a detailed description of their path-finding algorithms.

It is very common in the literature to use Equation (2.2) with a simple, usually constant, *bias* estimation for the edge weight and assume Dijkstra’s shortest path algorithm is used to find the best payment path [18]. This is a reasonable assumption and suitable for the purpose of our proposal. Note that the actual distribution of LN implementations in Bitcoin is clearly dominated by LND with more than a 90% share in 2022 according to [4], which might have decreased a bit nowadays, but it is still considered the dominant implementation by far. In this sense, assuming a single best path finding algorithm for the whole network seems reasonable, helping approximate realistic metrics. The relevant part is that these path-finding algorithms are deterministic. This determinism is what makes them predictable and thus prone to information disclosure as we will show. That is, looking for the *best* path at a given moment is deterministic. No matter what each implementation considers to be the *best*, it will always be the same.

2.3 Related work

Several works analyze different aspects related to the privacy of the LN [6,13]. Some works propose different approaches to deanonymize balances on the LN [5,6]. If an attacker can observe or estimate the balances, then multi-hop payment can be more easily deanonymized. These attacks usually require the adversary to actively perform actions with associated costs and are usually targeted to specific nodes.

The idea of deanonymizing multi-hop payments as considered in this paper has not been deeply explored, but is not entirely new. Most notably, [18] considers similar scenarios using shortest paths to evaluate the privacy of the payment in general payment network. Our proposal provides a more clear formalization of the metrics, focusing on the LN, using the payment amount and proposing the heuristic approach. In [7], authors also consider how to determine source and destination anonymity sets for specific implementations. As an alternative, timing attacks are considered in [15] to deanonymize multi-hop payments.

There are some approaches which can make the attack proposed here more difficult or impossible, one of such example is the proposal of payment splitting [19].

An interesting note can be give about multipath payments, which can actually increase the risk of this kind of attacks by increasing the probability of the adversary being part of the payment [12].

3 Adversarial model

We consider the scenario where one of the intermediary nodes in a multi-hop payment attempts to determine the source and destination of the payment. E.g., in Figure 2, we consider the adversary to be some node v_a , which takes part in a multi-hop payment from node v_1 to node v_t .

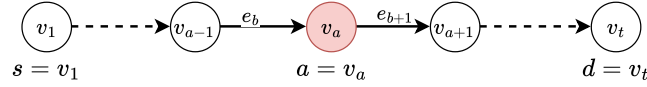


Fig. 2. Adversarial scenario.

Due to the use of onion routing, the adversarial node v_a , will know the preceding and following nodes, v_{a-1} and v_{a+1} , but not the rest of the nodes in the path. The adversary lies along the routing path, so it participates in two HTLCs: one with v_{a-1} and another with v_{a+1} . These HTLCs contain information regarding the payment transaction, most notably the amount x_{e_b} , that node v_a receives through the incoming edge e_b . Given a path, we will normally denote the source node as s , the destination as d , and the adversary as a .

What the attacker knows is that if $s = v_1$ is the sender node, and $d = v_t$ the destination, the payment path is a shortest path according to the path-finding algorithm, and such path must include the nodes v_{a-1} , v_a , v_{a+1} in this precise order. This idea will allow the adversary to discard some possible paths and even some possible sources and destinations.

4 Privacy metrics

We consider the anonymity sets associated to potential source and destination nodes, which will provide metrics on the privacy achieved in a concrete scenario determined by the network topology.

Given an adversary node v_a observing a payment, we denote as $\mathcal{S}$ the set of possible source nodes, and $\mathcal{D}$ the set of possible destination nodes. Initially, both sets will contain all nodes in the network except the attacker node. Note that the contiguous nodes v_{a-1} , and v_{a+1} could be source and destination respectively, so $\mathcal{S} = V \setminus \{v_a, v_{a+1}\}$, and $\mathcal{D} = V \setminus \{v_a, v_{a-1}\}$. $\mathcal{S}$, and $\mathcal{D}$ are the more general anonymity sets for the source and destination of the payment observed by the adversary v_a . The goal of the adversary is then to try to improve the estimation of both sets.

We will also consider probabilities associated to each node in $\mathcal{S}$, and $\mathcal{D}$, which will allow us to provide entropy based metrics [3,17,11,18]. Initially, all nodes could be considered to have the same probability, but we will show that these measures can be improved based on the existence of shortest paths. We denote as $P(s = v_i | a = v_a)$ the probability that v_i is the source node of a payment observed by attacker v_a , and $P(d = v_j | a = v_a)$ the probability of node v_j being the destination. In all cases we assume that the attacker is node $a = v_a$, which implies that the payment goes through channels e_b and e_{b+1} associated with v_a (see Figure 2).

We will denote the set of shortest paths between nodes u, v as $SP(u, v)$, and use $SP_{e_b, e_{b+1}}(u, v)$ for such shortest paths that include consecutively edges e_b , and e_{b+1} in the path.

4.1 Source and destination anonymity sets

The estimation of the source and destination anonymity sets is based on the fact that a payment between any two nodes will use the shortest path between such nodes. Here, as previously explained, by shortest we mean the best path according to the path-finding algorithm (see Section 2.2).

We based the anonymity sets in the property that a subpath of shortest path is a shortest path. We can define the set of all pairs of nodes (v_i, v_j) such that some shortest from v_i to v_j includes the edges e_b, e_{b+1} as:

$$SP_{e_b, e_{b+1}} = \{(v_i, v_j) \in V \times V \mid \exists P \in SP(v_i, v_j) \text{ such that } e_b, e_{b+1} \in P\}$$

Then, the source and destination anonymity sets can be defined in terms of the paths in $SP_{e_b, e_{b+1}}$.

Definition 1. *The source anonymity set for attacker node v_a , $\mathcal{S}_{v_a}$, is composed of all nodes $v_i \in V$, such that there exists a node u with shortest path from v_i to u passing through e_b and e_{b+1} .*

$$\mathcal{S}_{v_a} = \{v_i \mid \exists u \in V, (v_i, u) \in SP_{e_b, e_{b+1}}\}$$

We can analogously define the anonymity set for the destination node.

Definition 2. *The destination anonymity set for attacker node v_a , $\mathcal{D}_{v_a}$, is composed of all nodes $v_j \in V$, such that there exists a node u with shortest path from u to v_j passing through e_b and e_{b+1} .*

$$\mathcal{D}_{v_a} = \{v_j \mid \exists u \in V, (u, v_j) \in SP_{e_b, e_{b+1}}\}$$

The cardinality of both the source and destination sets are interesting metrics of the anonymity that the attacker has with respect to the source and destination of the payment.

In order for the adversary to evaluate shortest paths in the network it needs to estimate the amount used in the payment (see Section 2.1). This estimation can be done by the amount received at the attacker node x_a . The original amount, will be x_a minus the fees collected by nodes between the source node and the attacker. Given that fees in LN are relatively low, the adversary could assume x_a as the amount or provide a range to improve such estimation.

4.2 Probability based on shortest paths existence

We can associate a probability for each node of being source or destination of a payment, based on the number of shortest paths having such node as source or destination. The probability for node $v_i \in \mathcal{S}$ of being the source node of the payment seen by the attacker v_a , $P(s = v_i \mid a = v_a)$, can be estimated as:

$$P(s = v_i \mid a = v_a) = \frac{|\{u \in V \mid (v_i, u) \in SP_{e_b e_{b+1}}\}|}{|SP_{e_b e_{b+1}}|}$$

and analogously for nodes in $\mathcal{D}_{v_a}$, as $P(d = v_i \mid a = v_a)$. This allows us to determine an entropy-based metric for such sets as:

$$H(\mathcal{S}_{v_a}) = \frac{1}{\log_2 |SP_{e_b e_{b+1}}|} \sum_{v_i \in \mathcal{S}_{v_a}} Pr_{v_a}(s = v_i) \log_2 Pr_{v_a}(s = v_i)$$

and again, analogously for $H(\mathcal{D}_{v_a})$.

4.3 Heuristic based on rounded amount

To improve the destination anonymity set, we propose the use of an heuristic rule based on the idea that the payment is a rounded value. In this case, the adversary assumes that the amount of the payment is rounded, so the amount received by the destination node should be rounded. The adversary, can estimate the final payment received by nodes $v_i \in \mathcal{D}_{v_a}$, and then, discard those that do not represent a rounded payment.

What a rounded payment can be might be interpreted in different ways. Similarly to change detection heuristics [10], the value can be rounded to the currency used by the payment network, or a fiat currency such a dollars based on the change rate at the time of the payment.

5 Experiments

We have taken a snapshot of the Bitcoin Lightning Network from December 4th, 2025, using a CLN node. Details of the snapshot are shown in Table 1. From the snapshot we have taken only the active channels directions, which makes the number of directed edges to be not exactly the double of the channels. The mean capacity is 11,530,434 while the median is 2,475 satoshis.

Nodes	7,668
Directed edges	68,241
Channels	35,161

Table 1. LN snapshot characteristics

In order to simulate an adversarial attack we have taken a multi-hop payment between two random nodes, with a payment path of at least 3 nodes. The attacker is randomly chosen between the intermediate nodes. We have repeated this experiments 40 times for different amounts: 10, 100, 1000, 10000, 100000 satoshis. These amounts, using the exchange rate for December 4th, 2025 were equivalent to approximately 0.01, 0.1, 1, 10, and 100 EUR. We have intentionally tested relatively low values for the payments since the LN is precisely intended to be used for low value payments.

In order to simulate the payments, we have assumed a uniform balance distribution, where the capacity of each channel is uniformly distributed on each node (each direction).

5.1 Source and destination anonymity sets

Table 2 show the average cardinality of the source and destination anonymity sets, $|\mathcal{S}_{v_a}|$, and $|\mathcal{D}_{v_a}|$, and their respective entropy. For each value we provide also the coefficient of variation.

Amount	Source				Destination			
	$ \mathcal{S}_{v_a} $	CV	$H(\mathcal{S}_{v_a})$	CV	$ \mathcal{D}_{v_a} $	CV	$H(\mathcal{D}_{v_a})$	CV
10	438.525	2.181	4.40735	0.999	1157.45	1.917	5.97853	0.799
100	787.800	1.794	5.18258	0.919	657.525	1.963	5.02684	0.931
1000	573.325	2.480	4.52255	0.986	576.150	1.443	5.74129	0.780
10000	491.875	2.341	4.42918	0.960	691.425	1.995	4.91365	0.911
100000	320.850	2.752	4.42443	0.818	314.775	2.608	4.05264	0.967

Table 2. Source and destination anonymity set size and entropy.

The experimental results report the mean across 40 independent trials per payment amount. The coefficient of variation is observed to be large relative to the mean. Across all payment amounts, the CV for anonymity set sizes ranges from 1.4 to 2.7, indicating high relative variability. This dispersion is attributed to the heterogeneous topology of the Lightning Network, paths routed through highly connected hub nodes yield substantially larger anonymity sets, while peripheral paths produce considerably smaller ones. Notably, the CV for entropy measures remains stable at $0.78 \sim 1.0$ across all amounts. This shows that while a node’s position in the network influences the number of candidate nodes, the entropy-based metrics remain relatively stable, indicating that the adversary faces a consistent degree of uncertainty regardless of network position.

In some cases, the sets reported only one node, providing a complete deanonymization. As show in Table 3, in most cases, more than 30% of the times the source and anonymity cases contained only one element.

Amount	10	100	1000	10000	100000
$ \mathcal{S}_{v_a} = 1$	0.325	0.3	0.35	0.325	0.275
$ \mathcal{D}_{v_a} = 1$	0.325	0.325	0.3	0.325	0.3

Table 3. Ratio of cases for which the anonymity set contains one node.

5.2 Round heuristic

We have applied the round heuristic rounding to multiples of 100 msats, with an error tolerance of 5 msats. This allows us to obtain smaller destination anonymity destination sets with lower entropy as shown in Table 4.

Amount	Size	One node	Entropy	Success
10	639.275	0.3	5.07671	0.9
100	194.8	0.325	3.73994	0.85
1000	288.55	0.225	4.91227	0.825
10000	119.4	0.275	2.90912	0.525
100000	98.375	0.125	2.88617	0.425

Table 4. Application of the round heuristic.

Although the percentage of sets with one node is smaller, in general, the set sizes and entropy are much smaller.

It is important to note that the heuristic might fail. To show this we measure the success, as the percentage of times where the real destination node is actually in the destination anonymity set $v_d \in \mathcal{D}_{v_a}$. As shown in Table 4 this success decreases with the amount. Adjustments on the rounding multiple and error tolerance can increase this ratio, but will also increase the size and entropy of the anonymity set.

5.3 Discussion

Intermediate nodes in a multi-hop payment can get a relatively good estimation of source and destination nodes (recall the approximate 30% success deanonymizations). The relevant part is that this estimations can be done just by using publicly available information and without requiring active measures other than taking part in the payment. This honest but curious adversary does not require much effort.

It is important to revisit the assumptions underlying our experimental setup. The most critical assumption is that channel balances follow a uniform distribution. This provides a reasonable baseline for evaluation, but note that the adversary is assuming this uniform distribution, which happens to be the one used in the simulation. In practice, however, balance distributions are unlikely

to be uniform and are more plausibly bimodal. Under such conditions, an adversary relying on a uniform one would still obtain a reasonable approximation, although with increased estimation error particularly for high-value payments. Nevertheless, alternative estimations, such as a bimodal model, may introduce even larger errors due to misclassification: specifically, there is a 50% probability of incorrectly assigning the balance distribution for any given channel.

In general, for payments of low amount the actual distribution of balances might not be so important. Note that for the snapshot we have used the median capacity for channels is 2,475,000 sats, the mean is much bigger, 11,530,434 sats (possibly due to a few high capacity channels). This seems to suggest that small payments will likely succeed independently of the distribution of balances.

6 Conclusions

In this paper we have presented privacy metrics for lightning network multi-hop payment privacy. The metrics attempt to establish the source and destination of such payments and provide a realistic measure of the actual privacy achieved in the network. We have also considered the use of heuristic estimations to improve these metrics. As future work, we plan on introducing more heuristics and provide a more thoughtful empirical experimentation. This includes testing concrete implementation pathfinding algorithms and simulate a realistic balance distribution scenario.

Acknowledgement

This work is partially funded by the projects PID2021-125962OB-C33 SECURING/NET and PID2024-156914OB-C43 SAFE-BLOCKCHAIN, by the Ministerio de Ciencia e Innovación, the Agencia Estatal de Investigación and the European Regional Development Fund (ERDF).

References

1. Bitromortac. Blazing the trails: Improving lnd pathfinding reliability. Lightning Labs Blog <https://lightning.engineering/blog/>, 2024. Accessed: 2025-07-16.
2. G. Danezis and I. Goldberg. Sphinx: A Compact and Provably Secure Mix Format. In *2009 30th IEEE Symposium on Security and Privacy*, pages 269–282, May 2009.
3. Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Proceedings of the Privacy Enhancing Technologies Workshop (PET 2002)*, pages 54–68. Springer-Verlag, 2002.
4. @heisenberg. Proportion of lightning node implementations. Stacker News, <https://stacker.news/items/97320>, November 2022.
5. Jordi Herrera-Joancomartí, Guillermo Navarro-Arribas, Alejandro Ranchal-Pedrosa, Cristina Pérez-Solà, and Joaquin Garcia-Alfaro. On the difficulty of hiding the balance of lightning network channels. *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 602–612, 2019.

6. George Kappos, Haaroon Yousaf, Ania Piotrowska, Sanket Kanjalkar, Sergi Delgado-Segura, Andrew Miller, and Sarah Meiklejohn. An Empirical Analysis of Privacy in the Lightning Network. In *Financial Cryptography and Data Security*, pages 167–186, Berlin, Heidelberg, 2021. Springer.
7. Satwik Prabhu Kumble, Dick Epema, and Stefanie Roos. How lightning’s routing diminishes its anonymity. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, ARES ’21, pages 1–10, New York, NY, USA, August 2021. Association for Computing Machinery.
8. Satwik Prabhu Kumble and Stefanie Roos. Comparative analysis of lightning’s routing clients. In *2021 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pages 79–84, 2021.
9. Lightning Labs. Pathfinding: take channel capacity into account. lnd: Lightning Network Daemon Project, Comment on issue #5988 (by user bitromortac), <https://github.com/lightningnetwork/lnd/issues/5988#issuecomment-1131234858>, 2022.
10. Malte Möser and Arvind Narayanan. Resurrecting address clustering in bitcoin. In *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 386–403. Springer International Publishing, 2022.
11. Steven J. Murdoch. Quantifying and measuring anonymity. In *Data Privacy Management and Autonomous Spontaneous Security (DPM/SETOP 2013)*, pages 3–13. Springer, 2013.
12. Charmaine Ndolo and Florian Tschorsch. On the (not so) surprising impact of multi-path payments on performance and privacy in the lightning network. In *Computer Security. ESORICS 2023 International Workshops*, pages 411–427, Cham, 2024.
13. Rene Pickhardt, Sergei Tikhomirov, Alex Biryukov, and Mariusz Nowostawski. Security and Privacy of Lightning Network Payments with Uncertain Channel Balances. arXiv:2103.08576, March 2021.
14. Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. Technical report, Nakamoto Institute, 2016. Whitepaper; Accessed on May 6, 2025.
15. Elias Rohrer and Florian Tschorsch. Counting down thunder: Timig attacks on privacy in payment channel networks. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, AFT ’20, pages 214–227, New York, NY, USA, October 2020. Association for Computing Machinery.
16. Sindura Saraswathi and Christian Kümmerle. An exposition of pathfinding strategies within lightning network clients, 2025. arXiv preprint arXiv:2410.13784 [cs.NI].
17. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of the Privacy Enhancing Technologies Workshop (PET 2002)*, pages 41–53. Springer, April 2002.
18. Piyush Kumar Sharma, Devashish Gosain, and Claudia Diaz. On the Anonymity of Peer-To-Peer Network Anonymity Schemes Used by Cryptocurrencies. In *Proceedings 2023 Network and Distributed System Security Symposium*. Internet Society, 2023.
19. Gijs van Dam. Payment Splitting in Lightning Network as a Mitigation Against Balance Discovery Attacks, 2023.