Balance Disclosure in Payment Channel Networks using Noiseless Privacy

PADRAIG CORCORAN, School of Computer Science and Informatics, Cardiff University, UK
IRENA SPASIĆ, School of Computer Science and Informatics, Cardiff University, UK

Payment Channel Networks (PCNs) represent an emerging approach to improving the scalability of cryptocurrencies. A PCN provides the ability to consolidate a larger set of payments into a smaller set where individual payments are immediately confirmed. It consists of a network of payment channels between pairs of peers, where each channel has a total capacity that is split into two directional balances, determining the maximum payment amount that can be forwarded in either direction. If two peers do not have a direct channel between them, they can still make a corresponding payment using a path in the network where all channels in this path have a sufficient balance to forward the payment in question. Whenever a channel is used to make a payment, its balances are updated accordingly. To provide payment privacy, it is standard practice for channels not to disclose their current balances. However, this complicates the task of path planning, as a trial-and-error approach is often required in the search for a feasible path. In this article, we propose a novel method for disclosing channel balance information in a manner that both provides payment privacy and supports the task of path planning. This is achieved by applying methods from the field of Noiseless Privacy (NP) that allow balance summary statistics to be disclosed while obscuring the details of individual payments. Using a simulation of payments on the Lightning Network, which is built on top of the Bitcoin cryptocurrency, we demonstrate the trade-off between the level of privacy provided and the resulting effectiveness of path planning.

CCS Concepts: • **Mathematics of computing → Graph algorithms**; • **Networks → Network design and planning algorithms**; • **Computing methodologies → Distributed algorithms**.

Additional Key Words and Phrases: cryptocurrency; payment channel networks; path planning; edge computing

ACM Reference Format:
Padraig Corcoran and Irena Spasić. 2018. Balance Disclosure in Payment Channel Networks using Noiseless Privacy. *Distrib. Ledger Technol.* 37, 4, Article 111 (August 2018), 29 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

A cryptocurrency is a digital currency with popular examples including Bitcoin and Ethereum. Cryptocurrencies use blockchain technology to maintain a ledger of all past payment transactions. Due to the challenges of securely achieving decentralisation, cryptocurrencies typically exhibit scalability issues relating to low payment throughput (number of payments per second) and payment latency (time until a payment can be considered settled). For example, Bitcoin currently only processes approximately seven payments per second and has a payment latency of approximately ten minutes. The challenge of simultaneously achieving all three properties of decentralisation, security and scalability is commonly referred to as the blockchain trilemma.

Authors' Contact Information: Padraig Corcoran, corcoranp@cardiff.ac.uk, School of Computer Science and Informatics, Cardiff University, Cardiff, UK; Irena Spasić, spasici@cardiff.ac.uk, School of Computer Science and Informatics, Cardiff University, Cardiff, UK.

ACM 2769-6480/2018/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

Several potential solutions to improve the scalability of cryptocurrencies have been proposed [10]. In this work we focus on *payment channel networks* (PCNs), which are one of the most well-developed and promising solutions. An example of a PCN is the Lightning Network (LN) that is designed specifically for the Bitcoin cryptocurrency [19]. A PCN provides the ability to consolidate a larger set of payments into a smaller set, where individual payments are immediately confirmed. To demonstrate the operation of a PCN, consider the case where Alice and Bob wish to perform a series of payments between each other over a period of time. Instead of recording each individual payment in the blockchain ledger, Alice and Bob can create a shared payment channel that tracks their respective local balances. When Alice and Bob have completed their transacting, they can then settle the final local balances by recording them as a single payment in the blockchain ledger. By recording only a single payment in the blockchain, payment throughput is significantly increased. Furthermore, since each individual payment is only processed locally, the payment latency is also significantly reduced. If a network of payment channels exists, peers not sharing a channel can make payments between themselves using a sequence of channels that form a payment path in the network. For example, consider the case where Alice wants to make a payment to Charlie, but the parties in question do not share a channel. If Alice shares a channel with Bob, and Bob shares a channel with Charlie, then Alice can forward the payment to Bob, and Bob can, in turn, forward the payment to Charlie. To incentivise intermediate peers in payment paths to participate in forwarding payments, they are paid a fee that is proportional to the payment amount.

In order for a payment to be forwarded using a payment path, each channel in the path must have liquidity or balance in the payment direction that is greater than or equal to the payment amount. Furthermore, when a channel forwards a payment, the corresponding channel balances will update accordingly. In PCNs, channel balances are not disclosed or shared with other peers in the network for the following two reasons. First, the balance of a given channel will change each time that channel is used to make a payment. Therefore, communicating each change in a channel balance would generate a significant communication overhead. Second, by monitoring the precise timing and magnitude of balance changes across multiple channels, an adversary can correlate these events to reconstruct payment paths and infer payment details, including the sender, the recipient and the payment amount [12]. This neglects a principal goal of a PCN which is to provide payment privacy [11].

However, the policy of not disclosing channel balances presents several challenges. Firstly, the task of finding a feasible payment path for making a given payment becomes a trial-and-error process where a series of payment paths are computed and attempted until a feasible one with sufficient channel balances is found or the search is terminated [18]. Apart from adding additional computational complexity and payment latency, this process also makes those peers involved in any failed attempts aware of the existence of the payment in question [28]. Secondly, due to a lack of historical channel balance information, optimising the network topology and the distribution of liquidity to support payments is challenging. For example, Sivaraman et al. [24] demonstrated that, due to a poor distribution of liquidity, in many cases a given PCN may not contain any feasible payment paths for many payments. Furthermore, Kotzer and Rottenstreich [13] demonstrated the potential existence of Braess's paradox in a PCN whereby naively adding additional channels can reduce the efficiency of the network.

In this paper, we propose a novel method for disclosing channel balance information in the LN that provides payment privacy and supports path planning without the introduction of significant communication overhead. This is achieved through the use of methods from the field of Noiseless Privacy (NP) [3, 6]. In contrast to methods in the field of Differential Privacy (DP) [8] that achieve privacy by adding noise to the information disclosed, these methods achieve privacy by only disclosing summary statistics that have inherent uncertainty. In the context of the current problem, only the sum of several payment amounts is disclosed instead of individual payment amounts. Since there exist a vast number of distinct sets of payment amounts that sum to the same value, this introduces uncertainty regarding the individual payment amounts. Furthermore, since only the sum of several payment amounts is disclosed, communication overhead is minimised.

The remainder of this paper is structured as follows. In Section 2 we review related works on privacy and disclosing channel balance information. In Section 3 we describe the proposed method for disclosing LN channel balance information. This section also considers the potential of using DP methods to solve the problem in question and uses this as a motivation for considering NP methods. In Section 4 we present both a theoretical and experimental analysis of the proposed method where the latter analysis uses a simulation of synthetic payments on a real snapshot of the LN. Finally, in Section 5 we draw conclusions from this work and discuss possible directions for future research.

2 RELATED WORKS

In this section, we review related works on privacy and disclosing channel balance information in the LN. There are many potential methods that adversaries can use to infer information about payments. We now briefly review some of these methods. An adversary can use knowledge of changes in LN channel balances to infer payment information. In the absence of LN channels actively disclosing any balance information, which is currently the standard practice, changes in LN channel balances can be inferred by performing a probing attack [11, 27]. This attack exploits the fact that a given channel can successfully forward a payment if and only if it has a sufficient balance. Therefore, by attempting multiple fake payments of different amounts using a given channel, an adversary can estimate the balance in question. As mentioned in the introduction, knowledge of changes in channel balances can be used to infer all aspects of a payment. Rahimpour and Khabbazian [20] proposed a method that uses multi-path payments to improve the efficiency of probing attacks. Singh et al. [23] subsequently proposed a Bayesian method that further improves the efficiency of probing attacks. Dotan et al. [5] proposed a method to protect against probing attacks that involves channels randomly rejecting some payments even when they have a sufficient balance.

An adversary can use knowledge of payments processed by any LN channels it controls to infer payment information. For any payment that passes through such channels, the adversary will know approximately the payment amount (give or take the relatively small cost of routing fees). They will also know the channels in the payment path directly before and directly after the channels they control. It has been demonstrated that the source and destination of the payment can be estimated using this knowledge [12, 14]. Kappos et al. [12] proposed a simple heuristic that assumes the channels directly before and directly after the channel controlled are the first and last channels respectively in the payment path. Rohrer and Tschorsch [22] demonstrated how message timing information can be used to provide an even better estimate of a payment's source and destination. Nisslmueller et al. [16] proposed a method of adding time delays to protect against such attacks. van Dam and Kadir [29] proposed a method for hiding the true payment amount by adding noise to this value. This is implemented through the use of an additional circular payment back to the sender.

In the LN, when a sender wishes to perform a payment, it is generally necessary that they know the identity of the payment recipient. Blinded paths have been proposed to extend the LN protocol by allowing a payment recipient to avoid disclosing their identity to the sender [26]. Tang et al. [25] considered a method for disclosing channel balances that involve disclosing the true balance of a subset of channels after each payment. The authors demonstrated that this method did not preserve privacy, whereby the source and destination of a payment could be identified with high probability.

Pickhardt [17] proposed the development of a method where LN peers share perfectly accurate balance information with their "friends". The method proposed in this work can be considered a generalisation of this approach that involves sharing less accurate balance information with all peers in the network.

3 METHODOLOGY

In this section we propose a method for disclosing LN channel balance information. As indicated previously, this approach uses methods from the field of NP and, to motivate this approach, we also consider the use of methods from the field of DP. The remainder of this section is structured as follows. In Section 3.1 we more formally explain and define the problem of disclosing balance information. In Section 3.2 we demonstrate the challenges of using DP methods to solve this problem. Finally, in Section 3.3 we describe the proposed method based on NP.

3.1 Problem Definition

We model the LN as a directed graph $G = (V, E)$. Each peer in the network is modelled as an individual vertex $v \in V$. Each payment channel is modelled as a corresponding pair of directed edges, $(v_i, v_j) \in E$ and $(v_j, v_i) \in E$, between the same two vertices but pointing in opposite directions. A given channel will contain a total amount of liquidity that is distributed between the corresponding pair of edges. For a given channel, we refer to the total liquidity as its capacity. This capacity is distributed between the two corresponding edges, and the amount allocated to each edge is referred to as its balance. Consider the pair of edges $e \in E$ and $e' \in E$ corresponding to a given channel. We denote the corresponding balances as $b^e \in \mathbb{N}^0$ and $b^{e'} \in \mathbb{N}^0$ respectively. Furthermore, we denote the corresponding capacities as $c^e \in \mathbb{N}^+$ and $c^{e'} \in \mathbb{N}^+$ respectively. The following properties hold: $c^e = c^{e'}$, $b^e \in [0, c^e]$, $b^{e'} \in [0, c^{e'}]$ and $b^e + b^{e'} = c^e$.

Consider the pair of edges $e \in E$ and $e' \in E$ corresponding to a given channel. If the amount $a \in \mathbb{N}^+$ is transferred along the edge e, the balance b^e reduces by a while the balance $b^{e'}$ increases by a. For example, consider the case where $b^e = 5$ and $b^{e'} = 7$. If a payment amount 2 is forwarded using the edge e, the corresponding balances will become $b^e = 3$ and $b^{e'} = 9$. To transfer an amount a using an edge e, the condition $a \leq b^e$ must be satisfied. That is, the edge in question must have a sufficient balance.

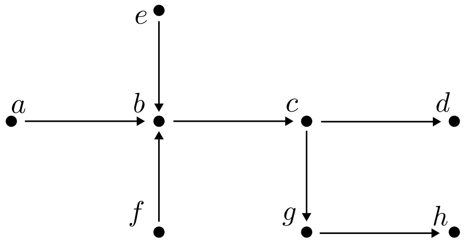
If two vertices do not share a common edge, payments between these vertices can still be made using a payment path. We define a payment path from $v_i \in V$ to $v_j \in V$ as a sequence of edges that form a path connecting the vertices in question. We refer to the number of edges in a payment path as the length of the path. The edges in a payment path charge a fee for forwarding a payment where the fee is subtracted from the payment amount forwarded. For the purpose of this work, it is not necessary to define the exact edge fee structure; however, the interested reader can find this information in [1]. We define the fee corresponding to a given payment path as the sum of all fees charged by the edges for forwarding the payment amount in question. A payment path can be used to forward a given payment if each edge in the path has a balance greater than or equal to the payment amount plus the fees charged by subsequent edges in the path. Finally, we define the path planning problem as the problem of determining a feasible minimum fee payment path to make a given payment [4].

For the reasons described above, only channel capacities and not edge balances are shared. Therefore, path planning is typically a trial-and-error process. A sender first estimates the balances along a potential path, often using heuristics or feedback from prior attempts. The payment is then sent along the calculated lowest-fee path. If it fails, feedback on the first edge with insufficient balance is used to update the estimates, and the process is repeated. These steps are repeated until the payment succeeds or the process is terminated. The above description of the path planning process implies that, more accurate edge balance estimates result in greater payment success using fewer payment attempts and the discovery of less expensive paths. In fact, if the true balances are precisely known and a feasible path exists, then a payment can be successfully made using a minimum fee path in a single attempt.

In this work we consider the research problem of disclosing edge balance information in a manner that both provides payment privacy and supports path planning. We define the *balance sequence* corresponding to a given edge as a sequence of balance and time pairs that model the changes in the edge balance over time. For example, consider the balance sequence $s^e = (b_0, t_0), (b_1, t_1), \dots, (b_m, t_m) \in ([0, c^e] \times \mathbb{R})^{\mathbb{N}^+}$ corresponding to the edge $e \in E$. In this example, b_0 equals the initial value of b^e when the edge e was first created and t_0 equals the time when this event occurred. Furthermore, b_1 equals the value of b^e after a single payment has been processed by the channel in question and t_1 equals the time when this event occurred. Finally, b_m equals the value of b^e after the most recent payment has been processed by the channel in question and t_m equals the time when this event occurred. The length $m + 1$ of a balance sequence may vary for different edges and will equal the number of payments m processed by the channel plus one. In this work we assume an adversary that continuously observes the LN and wishes to infer the set of balance sequences $\{s^e : e \in E\}$. Given this information, the adversary can infer the existence, value, source and destination of individual payments by clustering payment values and times that are implicit in the set of balance sequences.

To demonstrate how this inference can be performed, consider the example LN displayed in Figure 1(a). Although we model each channel with two edges pointing in opposite directions, to simplify understanding, this example only considers a subset of the edges. Figure 1(b) displays the set of balance sequences after all edges have been created but no payments have been made. For example, we can see that the edge (a, b) has a balance of 10 when it is initially created at time 0. Figure 1(c) displays the set of balance sequences after a single payment of value 3 has been made using the payment path (a, b), (b, c), (c, d) at time 2. Note that, in this example, we have assumed that the fees charged by all edges are zero and that all edges involved in a given payment update their respective balances simultaneously. The value, path and time of the above payment can all be directly inferred from the balance sequences by observing that all edges in the path simultaneously reduced their corresponding balances by the payment amount. Finally, Figure 1(d) displays the set of balance sequences after two additional payments have been made. The first is a payment of value 1 made using the payment path (e, b), (b, c), (c, g) at time 5 and the second is a payment of value 2 made using the payment path (f, b), (b, c), (c, g) at time 9. Again, the details of these payments can be directly inferred from the balance sequences.

In this work, we assume that LN channel operators will not coordinate when disclosing information about their respective edge balances. Hence, a decentralised solution is necessary. Given this, we formulate the problem of disclosing edge balance information in a manner that both provides payment privacy and supports path planning as follows. Define a function $f^e : ([0, c^e] \times \mathbb{R})^{\mathbb{N}^+} \rightarrow [0, c^e]$ for each $e \in E$ that maps the balance sequence s^e to a value that discloses information about the current balance b^e. Note that this formulation generalises to the case where only a subset of LN channels participate in the process of disclosing edge balance information. This is achieved by defining the functions f^e corresponding to those edges that do not participate to return a value unrelated to their respective balances. The set of functions $\{f^e | e \in E\}$ should have the following two properties. Firstly, a user who continuously observes the output of these functions should be able to use this information to efficiently perform the task of path planning. Secondly, an adversary who continuously observes the output of these functions should not be able to easily infer the



(a)

Edge ((v, v'))	Balance sequence ($s^{(v,v')}$)
(a, b)	$(10, 0)$
(b, c)	$(7, 0)$
(c, d)	$(3, 0)$
(e, b)	$(6, 0)$
(c, g)	$(8, 0)$
(f, b)	$(9, 0)$
(g, h)	$(4, 0)$

(b)

Edge ((v, v'))	Balance sequence ($s^{(v,v')}$)
(a, b)	$(10, 0), (7, 2)$
(b, c)	$(7, 0), (4, 2)$
(c, d)	$(3, 0), (0, 2)$
(e, b)	$(6, 0)$
(c, g)	$(8, 0)$
(f, b)	$(9, 0)$
(g, h)	$(4, 0)$

(c)

Edge ((v, v'))	Balance sequence ($s^{(v,v')}$)
(a, b)	$(10, 0), (7, 2)$
(b, c)	$(7, 0), (4, 2), (3, 5), (1, 9)$
(c, d)	$(3, 0), (0, 2)$
(e, b)	$(6, 0), (5, 5)$
(c, g)	$(8, 0), (7, 5), (5, 9)$
(f, b)	$(9, 0), (7, 9)$
(g, h)	$(4, 0)$

(d)

Fig. 1. An example LN network $G = (V, E)$ is displayed in (a) where vertices (V) and edges (E) are represented by circles and arrows respectively. The tables in (b), (c) and (d) represent the corresponding balance sequences ($s^{(v,v')}$ for $(v, v') \in E$) after zero, one and three payments respectively have been made.

specific details of individual payments. The details in question are the payment source, destination and amount. In addition, we assume an adversary may know the value of every payment forwarded by a subset of edges, which is equivalent to knowing their complete balance sequences. We denote this subset of edges as E^A. This would occur, for instance, if the adversary controls one of the vertices in such an edge.

Having defined the problem, we now evaluate the suitability of two privacy paradigms, Differential Privacy (DP) and Noiseless Privacy (NP), as potential solutions.

3.2 Differential Privacy

Differential Privacy (DP) is a paradigm for defining functions that disclose information about a dataset while preserving the privacy of individual data elements [8]. DP methods generally achieve this goal by adding sufficient noise to the function output such that a change in the function input dataset by a single data element cannot be detected. This approach assumes that an adversary has complete knowledge of the dataset apart from the single data element that changes. Consider the following function f^e that maps the balance sequence $s^e = (b_0, t_0), (b_1, t_1), \dots, (b_m, t_m)$ to b_m which equals the current balance b^e.

$$\begin{gathered} f^e : ([0, c^e] \times \mathbb{R})^{\mathbb{N}^+} \to [0, c^e] \\ (b_0, t_0), (b_1, t_1), \dots, (b_m, t_m) \mapsto b_m \end{gathered} \tag{1}$$

A party that continuously observes the output of the function f^e can easily infer s^e. In this section, we investigate whether a DP method that adds noise to the output of this function can prevent this attack while still supporting path planning. The remainder of this section is structured as follows. In Section 3.2.1 we provide a more formal introduction to DP and present a DP method known as the Laplace Mechanism. In Section 3.2.2 we consider the application of the Laplace Mechanism to the function presented above.

3.2.1 Background.

Broadly speaking, a function is DP if a change in the function's input dataset by a single data element does not cause a significant change in the function's output probability. Therefore, in the worst case scenario, when an adversary has complete knowledge of the dataset apart from the data element that changed, they cannot infer this particular element. Before formally defining DP, it is first necessary to define the concept of adjacent datasets.

Definition 1. *Let $\mathcal{X}$ be the space of datasets where a given dataset contains a set of data elements. A dataset $x \in \mathcal{X}$ is defined as being adjacent to a dataset $x' \in \mathcal{X}$ if and only if x and x' differ by a single data element. This relation is symmetric; that is, x is adjacent to x' if and only if x' is adjacent to x. The set of all datasets adjacent to a given dataset $x \in \mathcal{X}$ is denoted $adj(x)$.*

Note that the above definition does not formally define the criterion used to determine whether or not two datasets differ by a single data element. This criterion depends on the space of datasets $\mathcal{X}$ in question and the application or context in which it is used. In the context of the function defined in Equation 1, the space of datasets equals the space of balance sequences $([0, c^e] \times \mathbb{R})^{\mathbb{N}^+}$, and the data elements are individual payments. We define two balance sequences as being adjacent if one can be obtained from the other by the addition or removal of a single payment record. Given this, we formally define DP.

Definition 2. *Let $\mathcal{X}$ be the space of datasets. A randomised function $f : \mathcal{X} \longrightarrow \mathbb{R}$ provides (ε, δ)-DP if for all sets S in the range of f, and for all datasets x and x' in the domain of f that are adjacent, the following condition holds [8].*

$$Pr[f(x) \in S] \leq \exp(\varepsilon) Pr[f(x') \in S] + \delta. \tag{2}$$

In the above definition, from the perspective of an adversary, the function f is random and the probabilities are defined with respect to this function. The Laplace Mechanism is a method that takes a deterministic function and adds sufficient noise to its output so that it achieves (ε, δ)-DP for given values of ε and δ. Before defining this method, it is necessary to define the concepts of the sensitivity of a function and the Laplace probability distribution. Informally, the sensitivity of a function is a measure of the maximum possible change in the function output given the smallest possible change in the function input. It is defined as follows.

Definition 3. *Let $\mathcal{X}$ be the space of datasets and $f : \mathcal{X} \longrightarrow \mathbb{R}$ be a given function. The sensitivity of this function is denoted Δf and is defined as follows.*

$$\Delta f = \max_{\substack{x, x' \in \mathcal{X} \\ x' \in adj(x)}} |f(x) - f(x')|. \tag{3}$$

The Laplace distribution is a continuous probability distribution that is parametrised by a location $\mu \in \mathbb{R}$ and a scale $b \in \mathbb{R}$. It is a long-tailed distribution with a relatively large variance equal to $2b^2$. Given this, we formally define the Laplace Mechanism.

Definition 4. *Let $\mathcal{X}$ be the space of data and $f : \mathcal{X} \longrightarrow \mathbb{R}$ be a given function. The Laplace mechanism is defined as*

$$\mathcal{M}_L(x, f(.), \varepsilon) = f(x) + Y \tag{4}$$

where Y is a random variable drawn from a Laplace distribution with parameters $\mu = 0$ and $b = \Delta f / \varepsilon$.

It has been proven that the Laplace mechanism achieves $(\varepsilon, 0)$-DP (see Theorem 3.6 in [8]). Informally, the Laplace mechanism adds sufficient noise to the function output such that a change in this output that is less than or equal to the corresponding sensitivity, and in turn a change in the input dataset by a single data element, cannot be inferred with high probability. That is, one cannot determine whether such a change in the output is a consequence of a change in the input dataset or the added noise.

3.2.2 Disclosing Balance Information using DP.

In this section we consider whether applying the Laplace mechanism to the function f^e defined in Equation 1 provides a solution to the problem of providing payment privacy and supporting path planning. Towards this goal, in the following theorem we define the sensitivity of f^e denoted Δf^e:

Theorem 1. *The sensitivity Δf^e of the function f^e defined in Equation 1 equals the capacity c^e of the edge e.*

Proof. The function f^e returns the current balance b^e, which is always in the interval $[0, c^e]$. To determine the sensitivity, consider two adjacent balance sequences. Let s be a balance sequence where the final balance is $b_m = c^e$. Now consider an adjacent sequence s' formed by adding one more payment that transfers the entire balance. The new final balance will be $b_{m+1} = c^e - c^e = 0$. The difference in the function's output is $|f^e(s) - f^e(s')| = |c^e - 0| = c^e$. Since no single payment can change the balance by more than the total capacity, this represents the maximum possible change, and thus the sensitivity Δf^e equals c^e. □

As stated in the previous section, the Laplace mechanism adds a level of noise to the output of a function such that a change in this output that is less than or equal to the corresponding sensitivity cannot be inferred with high probability. The final balance value b_m in the balance sequence s^e will be a value in the interval $[0, c^e]$. This interval has a size of c^e, which also equals the sensitivity of the function f^e. Hence, the level of noise added to the function f^e will mean that no information about the value of b_m can be inferred apart from the fact that it lies in the interval $[0, c^e]$. In turn, this function will provide no additional balance information that can be used to support path planning.

Another potential challenge of applying the Laplace mechanism to the above function is that an observer may potentially draw multiple samples from the noisy output and use these to estimate the true edge balance (e.g. by computing the mean of the outputs). However, this challenge may be overcome using methods that limit the number of independent samples that can be drawn [5].

3.3 Noiseless Privacy

In the previous section we demonstrated the challenges of using DP methods to solve the problem of disclosing edge balance information. Similar to DP, Noiseless Privacy (NP) is a paradigm for defining functions that disclose information about a dataset while preserving the privacy of individual data elements. However, DP and NP differ in terms of how privacy is achieved. In DP, privacy is obtained by adding noise to the function that discloses information about the data. On the other hand, in NP, privacy is obtained by designing functions that disclose summary statistics that inherently have sufficient uncertainty and do not require noise to be added [3, 6, 9]. In this section we propose a novel method for disclosing edge balance information that uses NP methods.

In the proposed method, individual edges implement a policy where they infrequently disclose their current balance after multiple payments and not after each individual payment. For example, consider the case where an edge $e \in E$ implements a policy where they update the edge balance value disclosed after every n payments. This policy corresponds to the use of the following function

f^e to disclose edge balance information.

$$\begin{aligned} f^e : ([0, c^e] \times \mathbb{R})^{\mathbb{N}^+} &\longrightarrow [0, c^e] \\ (b_0, t_0), (b_1, t_1), \ldots, (b_m, t_m) &\longmapsto b_{m-(m \bmod n)}. \end{aligned} \tag{5}$$

For example, if $s^e = (10, 2), (7, 4), (5, 5)$ and $n = 3$, then $f^e(s^e) = 10$. Similarly, if $s^e = (10, 2), (7, 4), (5, 5), (8, 6)$ and $n = 3$, then $f^e(s^e) = 8$. Continuously observing the value of f^e would not disclose the balance sequence $s^e = (b_0, t_0), (b_1, t_1), \ldots, (b_m, t_m)$. Instead, the subsequence $s^e_n = (b_0, t_0), (b_{1n}, t_{1n}), (b_{2n}, t_{2n}), \ldots, (b_{\lfloor m/n \rfloor n}, t_{\lfloor m/n \rfloor n})$, where $\lfloor . \rfloor$ denotes the floor operation, would be disclosed. Hence, using the function f^e to disclose edge balance information is equivalent to disclosing s^e_n. We will later show that this is, in turn, equivalent to disclosing summary statistics about payment values and a subset of payment times.

To demonstrate the above method consider the example LN displayed in Figure 2(a). Figure 2(b) displays the set of balance subsequences after all edges have been created but no payments have been made. Figure 2(c) displays the set of balance subsequences after a single payment of value 3 has been made using the path $(a, b), (b, c), (c, d)$ at time 2. Note that since all edges have a disclosure frequency of $n = 2$, this first payment does not trigger any updates to the disclosed subsequences. Finally, Figure 2(d) displays the set of balance subsequences after two additional payments have been made. The first is a payment of value 1 made using the path $(e, b), (b, c), (c, g)$ at time 5 and the second is a payment of value 2 made using the path $(f, b), (b, c), (c, g)$ at time 9. Note that this is the same example LN and set of payments presented in Section 3.1 and Figure 1 when we demonstrated how one can infer payment details by clustering payment values and times. By examining the above balance subsequences, one can see that performing such clustering and, in turn, inference is more difficult. Note that, in the above example, the value of n was constant across time and edges. This is not a requirement and, as we will see later, this value can vary across time and edges.

The remainder of this section is structured as follows. In Section 3.3.1 we provide a formal introduction to NP. In Section 3.3.2, we prove that disclosing edge balance information using the function f^e defined in Equation 5 provides payment privacy and supports path planning. In doing so, we also present a method for computing a suitable value for the parameter n. In Section 3.3.3, we use the above results to define a policy for disclosing edge balance information in a manner that provides privacy. Finally, in Section 3.3.4 we present analysis demonstrating that disclosing edge balance information can improve the ability to perform path planning.

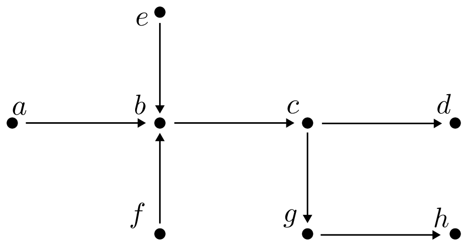
3.3.1 Background.

In NP we assume that an adversary wishes to infer the data elements in an unknown dataset. Given this, the goal is to exploit this uncertainty with respect to the dataset by only disclosing summary statistics (e.g. the sum of the data elements) from which the adversary cannot infer individual data elements. This approach is distinct from DP, which assumes that an adversary has complete knowledge of the dataset apart from a single data element. In the following we present several background definitions that form a foundation for subsequent analysis.

We now formally define NP by adopting the definitions proposed by Bhaskar et al. [3] and Grining and Klonowski [9]. Towards this, it is necessary to define the concept of sensitivity of a function and dataset. Informally, this is a measure of the maximum possible change in the output of a function applied to a dataset given the smallest possible change in this dataset.

Definition 5. *Let $x \in \mathcal{X}$ be a given dataset where $\mathcal{X}$ is the space of datasets. Let $f : \mathcal{X} \longrightarrow \mathbb{R}$ be a given function. The sensitivity of this function and dataset is denoted $\Delta f(x)$ and is defined as follows.*

$$\Delta f(x) = \max_{x' \in adj(x)} |f(x) - f(x')|. \tag{6}$$



(a)

Edge $((v, v'))$	n	Balance subseq $(s_n^{(v,v')})$
(a, b)	2	$(10, 0)$
(b, c)	2	$(7, 0)$
(c, d)	2	$(3, 0)$
(e, b)	2	$(6, 0)$
(c, g)	2	$(8, 0)$
(f, b)	2	$(9, 0)$
(g, h)	2	$(4, 0)$

(b)

Edge $((v, v'))$	n	Balance subseq $(s_n^{(v,v')})$
(a, b)	2	$(10, 0)$
(b, c)	2	$(7, 0)$
(c, d)	2	$(3, 0)$
(e, b)	2	$(6, 0)$
(c, g)	2	$(8, 0)$
(f, b)	2	$(9, 0)$
(g, h)	2	$(4, 0)$

(c)

Edge $((v, v'))$	n	Balance subseq $(s_n^{(v,v')})$
(a, b)	2	$(10, 0)$
(b, c)	2	$(7, 0), (3, 5)$
(c, d)	2	$(3, 0)$
(e, b)	2	$(6, 0)$
(c, g)	2	$(8, 0), (5, 9)$
(f, b)	2	$(9, 0)$
(g, h)	2	$(4, 0)$

(d)

Fig. 2. An example LN network $G = (V, E)$ is displayed in (a) where vertices (V) and edges (E) are represented by circles and arrows respectively. The tables in (b), (c) and (d) represent the corresponding balance subsequences ($s_n^{(v,v')}$ for $(v, v') \in E$) after zero, one and three payments respectively have been made.

Note that this definition of sensitivity is specific to both a given function and a given data. This is distinct from definition of sensitivity used in DP which is specific to only a given function. Finally, we formally define the concept of NP.

Definition 6. *Let $x \in \mathcal{X}$ be a random dataset where $\mathcal{X}$ is the space of datasets. Let $f : \mathcal{X} \longrightarrow \mathbb{R}$ be a given deterministic function. This function and dataset provide (ε, δ)-NP if for all sets S in the range of f and for all random datasets x' adjacent to x, the following condition holds.*

$$Pr[f(x) \in S] \leq \exp(\varepsilon)Pr[f(x') \in S] + \delta. \tag{7}$$

In the above definition, from the perspective of an adversary, the datasets x and x' are random and the probabilities are defined with respect to these datasets. Informally, the above definition models the change in the adversary's belief about a particular output in the range of f given a change in the input dataset by a single data element [3]. This has similarities to the definition of DP in Definition 2. If we contrast their respective definitions, we see that they differ with respect to where the adversary uncertainty originates and in turn their ability to infer data elements. In DP, adversarial uncertainty originates from the randomness of the function, while in NP, it originates from the randomness of the data itself.

3.3.2 Disclosing Balance Information using NP.

As previously demonstrated, disclosing edge balance information using the function f^e defined in Equation 5 is equivalent to disclosing the balance subsequence s_n^e. We will now show that this is, in turn, equivalent to disclosing the initial edge balance before any payments have been made, $\lfloor m/n \rfloor$ sums of payment values and $\lfloor m/n \rfloor$ payment times.

Consider the balance subsequence $s_n^e = (b_0, t_0), (b_{1n}, t_{1n}), (b_{2n}, t_{2n}), \dots, (b_{\lfloor m/n \rfloor n}, t_{\lfloor m/n \rfloor n})$. The first element (b_0, t_0) in this sequence equals the initial edge balance and time before any payments have been made. Hence, disclosing this element does not disclose any payment information. Each of the remaining elements (b_{in}, t_{in}) for $i = 1, \dots, \lfloor m/n \rfloor$ can be modelled as a pair of function outputs $f_{i,n}^{e,b}$ and $f_{i,n}^{e,t}$. The function $f_{i,n}^{e,b}$ maps the original balance sequence s^e to b_{in} and is defined as follows.

$$\begin{aligned} f_{i,n}^{e,b} &: ([0, c^e] \times \mathbb{R})^{\mathbb{N}^+} \to [0, c^e] \\ (b_0, t_0), (b_1, t_1), \dots, (b_m, t_m) &\mapsto b_{in} \\ &= b_{(i-1)n} + \sum_{j=0}^{n-1} \left(b_{(i-1)n+j+1} - b_{(i-1)n+j} \right) \end{aligned} \tag{8}$$

The term $b_{(i-1)n}$ in the above equation equals the output of the function $f_{i-1,n}^{e,b}(s^e)$. Each term $b_{(i-1)n+j+1} - b_{(i-1)n+j}$ in the above summation equals an individual payment value. This value is positive if the payment is made using the edge in question. This value is negative if the payment is made using the other edge corresponding to the channel in question. Hence, disclosing the terms b_{in} for $i = 1, \dots, \lfloor m/n \rfloor$ is equivalent to disclosing $\lfloor m/n \rfloor$ sums of n payment values. Note that the terms in each of these sums form independent (non-intersecting) sets.

The function $f_{i,n}^{e,t}$ maps the original balance sequence s^e to t_{in} and is defined as follows.

$$\begin{aligned} f_{i,n}^{e,t} &: ([0, c^e] \times \mathbb{R})^{\mathbb{N}^+} \to \mathbb{R} \\ (b_0, t_0), (b_1, t_1), \dots, (b_m, t_m) &\mapsto t_{in}. \end{aligned} \tag{9}$$

Hence, disclosing the terms t_{in} for $i = 1, \dots, \lfloor m/n \rfloor$ is equivalent to disclosing the payment times for $\lfloor m/n \rfloor$ payments.

In summary, disclosing edge balance information using the function f^e defined in Equation 5 is equivalent to disclosing the functions $f_{i,n}^{e,b}$ and $f_{i,n}^{e,t}$ for $i = 1, \dots, \lfloor m/n \rfloor$ defined in Equations 8 and 9 respectively. Therefore, any statement that can be proven for the latter functions also holds for the function f^e. In the following, we first prove that the set of functions $f_{i,n}^{e,b}$ achieve NP and define a method for selecting the parameter n. Since NP is achieved, this implies that reliably clustering payment paths using payment values is computationally difficult. We subsequently prove that the set of functions $f_{i,n}^{e,t}$ do not achieve NP. However, we demonstrate that the information disclosed by these functions is very limited making the task of clustering payment paths extremely difficult. Finally, we analyse potential information leakage between these disclosed functions.

Disclosure of Balance Functions.

First consider the functions $f_{i,n}^{e,b}$ for $i = 1, \dots, \lfloor m/n \rfloor$. Recall that each of these functions discloses a sum of n payment values. Given a dataset, Grining and Klonowski [9] presented methods for computing the (ε, δ)-NP parameter values for disclosing a sum of the data elements in question. If these computed parameter values are sufficiently small, one can infer that it is safe to disclose the sum in question. Otherwise, one should consider not disclosing the sum in question. The authors present several methods for computing these parameter values that make different assumptions regarding the dataset and level of adversary knowledge. We present one of these methods in Theorem 2, which is a rephrasing of Theorem 4 in the original article by Grining and Klonowski [9]. This theorem makes the following three assumptions:

(1) The data elements are independent. In the current context where data elements correspond to payment values and payments are made by a diverse set of users to pay for a diverse set of items, this is a reasonable assumption to make.

(2) The adversary knows the distribution of the data elements. In the current context where data elements correspond to payment values, this could occur if the adversary has some historical payment data, potentially from a different payment system, for estimating the distribution in question.
(3) The adversary knows the values of a subset of the data elements.

Theorem 2. *Let $X = (X_1, X_2, \ldots, X_n) \in \mathcal{X}$ be a given dataset where each data element X_i is an independent random variable. Assume that an adversary knows the values of a subset of X. Let Γ denote the indexes of this subset and let γ denote the relative size of this subset; that is $|\Gamma| = \gamma n$. Let $\mu_i = \mathbb{E}(X_i)$, $\sigma_\Gamma^2 = \frac{\sum_{i \in [n] \setminus \Gamma} \mathrm{Var}(X_i)}{(1-\gamma)n}$ and $\mathbb{E}(|X_i|^3) < \infty$ where $[n] = \{1, \ldots, n\}$ for $i \in \{1, \ldots, n\}$. Consider a function $f(X) = \sum_{i=1}^{n} X_i$. Let Δ denote the sensitivity of dataset X and function f. Given this, $f(X)$ provides (ε, δ)-NP with parameter values*

$$\varepsilon = \sqrt{\frac{\Delta^2 \ln((1-\gamma)n)}{(1-\gamma)n\sigma_\Gamma^2}} \tag{10}$$

and

$$\delta = \frac{1.12 \sum_{i \in [n] \setminus \Gamma} \mathbb{E}(|X_i - \mu_i|^3)}{\left(\sum_{i \in [n] \setminus \Gamma} \mathrm{Var}(X_i)\right)^{\frac{3}{2}}} (1 + e^{\varepsilon}) + \frac{4}{5\sqrt{(1-\gamma)n}} \tag{11}$$

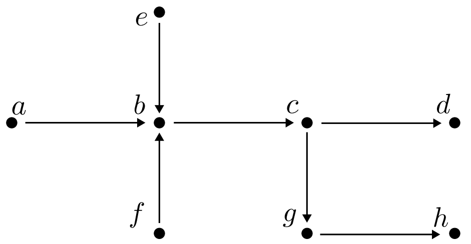
Proof. Please see Theorem 4 in the original article by Grining and Klonowski [9] for a proof of the above result. The proof method relies on the Berry-Esseen theorem to approximate the sum of the data elements with indexes in the set $[n] \setminus \Gamma$ with a Gaussian distribution. This approximation and its quantifiable error are then combined with properties of the Gaussian distribution to derive the final privacy parameters. □

The parameter values in Theorem 2 are defined with respect to the sensitivity of the dataset and function in question. In the context of a function $f_{i,n}^{e,b}$, the sum in question equals a sum of the values of payments made using both edges corresponding to the channel in question. Therefore, the sensitivity will equal the maximum possible payment that can be made using these edges, which equals the capacity of the channel in question.

The theorem's third assumption, that an adversary knows the values of a subset of the data elements, can arise in several practical scenarios. For instance, an adversary will directly know the value of any payment they initiate that is forwarded by the edge in question. Knowledge can also be gained from network topology. For example, if the edge in question is adjacent to another edge that discloses its balance after every payment, an adversary can infer the values of all payments forwarded by that adjacent edge. This information could then be used to determine if those same payments are subsequently forwarded by the edge in question. Finally, this assumption is consistent with our broader adversary model from Section 3.1, where we assume an adversary has knowledge of all payments forwarded by a subset of edges E^A, which could be used to deduce the values of payments on other edges within the same payment paths.

Disclosure of Time Functions.
Next consider the functions $f_{i,n}^{e,t}$ for $i = 1, \ldots, \lfloor m/n \rfloor$. Recall that, each of these functions discloses the time value for a single payment. In Theorem 3, we prove that each of these functions does not achieve NP.

Theorem 3. *Each of the functions $f_{i,n}^{e,t}$ defined in Equation 9 for $i = 1, \ldots, \lfloor m/n \rfloor$, does not achieve $(\varepsilon, 0)$-NP for any value of ε.*



(a)

Edge $((v, v'))$	n	Balance subseq $(s_n^{(v,v')})$
(a, b)	∞	$(10, 0)$
(b, c)	2	$(7, 0), (3, 6)$
(c, d)	∞	$(3, 0)$
(e, b)	∞	$(6, 0)$
(c, g)	2	$(8, 0), (5, 6)$
(f, b)	4	$(9, 0)$
(g, h)	∞	$(4, 0)$

(b)

Fig. 3. An example LN network $G = (V, E)$ is displayed in (a) where vertices (V) and edges (E) are represented by circles and arrows respectively. The table in (b) represents the corresponding balance subsequences ($s_n^{(v,v')}$ for $(v, v') \in E$) after four payments have been made.

Proof. For each of these functions, let $x = (b_0, t_0), \ldots, (b_{in}, t_{in}), \ldots, (b_m, t_m)$ and $x' = (b_0, t_0), \ldots, (b'_{in}, t'_{in}), \ldots, (b_m, t_m)$ be two adjacent random balance sequences where the payments (b_{in}, t_{in}) and (b'_{in}, t'_{in}) are different and $t_{in} \neq t'_{in}$. The values of t_{in} and t'_{in} are disclosed by the function $f_{i,n}^{e,t}$ and therefore are known by the adversary. Let us denote these values as a and b respectively. Hence, $Pr[f_{i,n}^{e,t}(x) \in \{a\}] = 1$ and $Pr[f_{i,n}^{e,t}(x') \in \{b\}] = 1$. In turn, $Pr[f_{i,n}^{e,t}(x') \in \{a\}] = 0$. Substituting $Pr[f_{i,n}^{e,t}(x) \in \{a\}]$ and $Pr[f_{i,n}^{e,t}(x') \in \{a\}]$ into Equation 7 we see that $(\varepsilon, 0)$-NP cannot be achieved for any value of ε. □

The above negative result is a consequence of the fact that $f_{i,n}^{e,t}$ discloses the exact value of t_{in} for $i = 1, \ldots, \lfloor m/n \rfloor$. Therefore, from the adversary's perspective, there is no uncertainty with respect to these values. However, if we assume that individual payments are independent, each function $f_{i,n}^{e,t}$ does not disclose any information about (b_j, t_j) for $j \neq in$. We argue that, if most edges use a value of n that is sufficiently large, the loss in privacy is relatively small.

Recall from Section 3.1, that to infer the details of a given payment, an adversary must perform a clustering of payment values and times. If a given edge uses the above method to update its edge balance every n payments, then only the corresponding time of one in every n payments processed by the edge will be disclosed. Assuming that edges do not synchronise to disclose their balances immediately after a given payment plus the presence of concurrent payments, we argue that, determining a correct clustering is non-trivial.

To demonstrate this argument consider the example LN displayed in Figure 3(a). Figure 3(b) displays the set of edge balance subsequences after the following four payments have been made: a payment of value 3 using the path $(a, b), (b, c), (c, d)$ at time 2, a payment of value 2 using the path $(c, g), (g, h)$ at time 4, a payment of value 1 using the path $(a, b), (b, c), (c, d)$ at time 6, and a payment of value 1 using the path $(c, g), (g, h)$ at time 6. Note that the values of n used by the edges are not all equal. In fact, four edges used a value of n equal to ∞. These edges only disclose their initial balance and can be considered as not participating. Examining the edge balance subsequences we can see that the final two payments that happened at time 6 caused the two edges (b, c) and (c, g) to disclose their current balances. Therefore, an adversary can infer that both edges processed a payment at time 6. However, clustering these edges based on this fact suggests that a payment was made from b to g, which is incorrect.

In the following we formalise the above argument. Recall that E^A denotes the subset of edges for which an adversary knows the corresponding balance sequences. For any $e \in E^A$, the adversary observes an updated balance immediately after any payment traverses it. For the remaining edges $E^N = E \setminus E^A$, from the adversary's perspective balance updates are disclosed sporadically, governed by some non-deterministic process (e.g., every n-th payment where payments are random).

At a discrete time t, a set of k concurrent payments $\mathcal{P}_t = \{P_1, P_2, \dots, P_k\}$ occur where each payment P_i has a corresponding payment path π_i. The adversary does not observe $\mathcal{P}_t$ directly. Instead, their knowledge is strictly the set of edges that both forwarded a payment and disclosed an update at time t. Formally, if we let $\Pi_t = \cup_{i=1}^{k} \pi_i$ be the set of all edges involved in payments at time t, and $D(e,t)$ be a predicate that is true if edge e discloses at time t, then

$$O_t = \{e \in \Pi_t \mid D(e,t)\} \tag{12}$$

This set of observed edges can be partitioned into two disjoint sets

(1) Anchor Disclosures: $O_t^A = O_t \cap E^A$.
(2) Floating Disclosures: $O_t^N = O_t \cap E^N$.

The adversary's objective is to partition the observed set O_t into clusters $\{C_1, \dots, C_k\}$ such that each cluster C_j consists of the complete set of observed disclosures for a single, unique payment path π_j (i.e., to find $C_j = \pi_j \cap O_t$ for each payment). The true path π_j may contain additional edges not present in C_j.

The set O_t^A provides the adversary with a high-confidence structural foundation. By partitioning this set based on topological proximity, known routing patterns and payment values, the adversary can infer the existence of k' payment path fragments; we define a payment path fragment as a high-confidence cluster of co-occurring anchor disclosures that the adversary hypothesizes as belonging to a single, unique payment path. To model the adversary's best-case scenario, we assume they correctly deduce the true number of payments, $k' = k$, and identify k corresponding anchor sets $\{A_1, \dots A_k\}$, where each set $A_j = O_t^A \cap \pi_j$ corresponds to the anchor disclosures from a single true payment path π_j.

Given the k anchor sets, the adversary's primary task is to assign the set of $h = |O_t^N|$ floating disclosures to these fragments. Formally, the adversary must find the correct partition $\{O_t^N(1), \dots, O_t^N(k)\}$ of O_t^N, where each subset $O_t^N(j) = O_t^N \cap \pi_j$ contains the floating disclosures belonging to the true payment path π_j. This is a combinatorial problem of partitioning the set O_t^N and assigning each element to one of the k distinct anchor sets. The number of ways to partition a set of h distinct elements into exactly k non-empty, labelled sets is given by $k! \times S_2(h,k)$ where $S_2(h,k)$ is the Stirling number of the second kind. The total number of possible assignments, which we denote $\mathcal{N}(h,k)$, is therefore

$$\mathcal{N}(h,k) = k! \times S_2(h,k) = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^h. \tag{13}$$

The value of $\mathcal{N}(h,k)$ grows exponentially with h (the number of floating disclosures), creating a large search space of plausible clusterings.

It should be noted that a sophisticated adversary would not treat all $\mathcal{N}(h,k)$ possible assignments as equally probable. They would employ heuristics to construct a probability distribution over the space of possible partitions. Possible heuristics include topological proximity and historical payment data. The adversary's task is thus reduced from a brute-force search to identifying the maximum-likelihood partition.

Crucially, the reliability of the aforementioned heuristics is fundamentally compromised by information loss regarding non-disclosing edges that do not disclose their current balance at the time t. Let U_t^N be the set of edges in E^N that forwarded a payment at time t but did not disclose an update.

$$U_t^N = \left(\bigcup_{i=1}^{k} (\pi_i \cap E^N) \setminus O_t^N \right) \tag{14}$$

The existence of this non-empty set U_t^N makes the problem intractable by undermining the adversary's heuristics:

(1) Path Incompleteness: Even if the adversary makes a correct assignment and forms the cluster $C_j = A_j \cup O_t^N(j)$, this cluster represents an incomplete path skeleton. The true path is $\pi_j = C_j \cup U_t^N(j)$ where $U_t^N(j)$ is the (potentially non-empty) set of non-disclosing edges from that path, about which the adversary knows nothing.
(2) Topological Ambiguity: The heuristic of topological proximity fails because the adversary cannot measure true path distance. An observed edge that appears distant may be connected by a short chain of on-disclosing edges from $U_t^N(j)$, rendering the heuristic unreliable.

The above analysis implies that the task of clustering payment edges is computationally hard due to a dual-layered problem. First, the adversary faces the challenge of optimizing over an exponential search space of possible assignments. Second, the very heuristics required to navigate this space are rendered unreliable by the incomplete and ambiguous nature of the observational data - a direct consequence of the balance disclosure mechanism. The adversary is thus left to solve an intractable assignment problem with incomplete information, making the reliable reconstruction of payment paths infeasible.

Analysis of Information Leakage Between Disclosed Functions.
In the proposed method, the functions $f_{i,n}^{e,b}$ and $f_{i,n}^{e,t}$ are disclosed jointly as pairs, providing both balance sums and corresponding disclosure times. While the privacy guarantees for each set of functions have been analysed separately, it is essential to examine their composition. In privacy frameworks such as Differential Privacy (DP), releasing multiple statistics can degrade overall guarantees through additive privacy budgets or information leakage [8]. Similarly, in Noiseless Privacy (NP), although privacy stems from inherent data uncertainty rather than added noise, joint disclosures could potentially interact if one reveals information that reduces uncertainty in the other. Therefore, before defining the disclosure policy, we must verify whether disclosing one set weakens the privacy of the other, considering both directions.

First, consider whether disclosing the balance sums $f_{i,n}^{e,b}$ weakens the privacy of the times $f_{i,n}^{e,t}$. The privacy of $f_{i,n}^{e,t}$ relies on the sparsity of disclosed times and the computational intractability of clustering edges into payment paths, as detailed above. This hardness arises from incomplete observations (U_t^N) and the exponential assignment space $\mathcal{N}(h, k)$, which depend on temporal and topological factors. The balance sums provide no additional constraints on timing or path structure, as payment amounts and times are independent. Thus, knowledge of sums does not refine clustering heuristics (e.g., topological proximity) or reduce the search space, preserving the privacy of $f_{i,n}^{e,t}$.

Conversely, consider whether disclosing the times $f_{i,n}^{e,t}$ weakens the privacy of the balance sums $f_{i,n}^{e,b}$. The privacy of $f_{i,n}^{e,b}$ is based on the uncertainty in decomposing each sum into n individual payment values, with guarantees from Theorem 2 assuming an adversary knows n, the payment distribution, and a fraction γ of values. An adversary could use the time interval $\Delta t = f_{i,n}^{e,t} - f_{i-1,n}^{e,t}$ to estimate the number of payments n, for instance, by modelling arrivals as a Poisson process with rate λ, yielding $\hat{n} \approx \lambda \Delta t$. However, this does not weaken the NP guarantee: the theorem is conservative and already assumes knowledge of n. An accurate estimate matches this worst-case scenario, while an inaccurate one increases decomposition uncertainty, strengthening privacy. Since amounts and times are independent, no further leakage occurs. In summary, the joint disclosure does not weaken the individual guarantees due to orthogonal uncertainty sources and independent data dimensions. This supports the robustness of the NP framework.

3.3.3 *Policy for Disclosure of Edge Balances.*

Theorem 2 provides a tool that can be used by edges to define a policy for disclosing edge balance information in a manner that provides privacy. Consider an edge and the case where this edge has made n payments since it last disclosed its current balance. Using this theorem, the edge can determine if n is sufficiently large to achieve payment privacy. If n is sufficiently large, the edge can update the value it discloses to equal the current balance. If n is not sufficiently large, the edge can instead continue to make payments until this condition is met.

The above policy is described more formally in Algorithm 1 using pseudocode. The algorithm takes as input the privacy thresholds ε^t and δ^t. An edge operator can use Theorem 2 to assist in selecting appropriate values for these thresholds, as the theorem provides the analytical tools to evaluate the privacy guarantees for a given number of aggregated payments. These threshold parameters are used to ensure that n is sufficiently large to achieve a desired level of privacy. The algorithm first initialises the list x of payment values to equal the empty list (line 1) and the balance value d currently disclosed to equal the true balance b^e (line 2). For each new payment new payment processed by the channel containing edge e, the following actions are performed. The algorithm adds the payment value to the list x (line 4); this value is positive for a payment of amount a forwarded by edge e and negative ($-a$) for a payment forwarded by the other edge in the same channel. It then computes the number of elements in x (line 5). It computes the values of ε and δ with respect to x using Equations 10 and 11 respectively (line 6). If the conditions $\varepsilon \leq \varepsilon^t$ and $\delta \leq \delta^t$ are satisfied (line 7), the algorithm resets x to equal the empty list and updates the disclosed balance d to equal the true balance b^e. Finally, this algorithm does not introduce a large communication overhead since each channel only communicates an edge balance update after a series of payments.

Algorithm 1: Policy used by edge $e \in E$ for disclosing balance subsequence.

```
  Input: Privacy thresholds $\varepsilon^t$ and $\delta^t$.
1 Initialise $x = [\,]$.
2 Initialise $d = b^e$.
3 while True do
4     $x$.append(next_payment_value()).
5     $n = |x|$.
6     Compute $\varepsilon$ and $\delta$ using Equations 10 and 11 respectively.
7     if $\varepsilon \leq \varepsilon^t$ and $\delta \leq \delta^t$ then
8         $d = b^e$.
9         $x = [\,]$.
```

Several practical challenges must be considered when integrating the proposed policy into the existing LN implementation. One approach that would not require changes to the base protocol is to implement the policy as an opt-in overlay network. In this model, participating vertices would broadcast their infrequent balance disclosures over a dedicated gossip network. This creates a marketplace for balance information with distinct incentives for participation.

The primary incentive for a vertex to publish its edge balance information is to increase its routing fee revenue; by providing more accurate data, its edges are more likely to be selected to be elements of payment paths. The primary incentive for a vertex to subscribe to this information is to improve its own payment outcomes by increasing success rates and reducing fees, as access to more accurate data allows for more effective path planning.

However, this approach introduces its own challenges. Firstly, at scale, the message overhead could become significant, likely requiring vertices to subscribe to only a curated subset of high-traffic edges. Secondly, the act of subscribing to an edge's updates is itself a metadata signal that could leak a vertex's path planning intentions to an adversary. Addressing these concerns represents an additional research challenge beyond the scope of the current work.

3.3.4 Path Planning using NP Balance Information.

As described in Section 3.1, a payment path can be used to forward a given payment if each edge in the path has a balance greater than or equal to the payment amount plus any fees charged by subsequent edges. Given that edge balances are not disclosed, path planning will typically be performed using a trial-and-error process where a series of payment paths are computed and attempted until a feasible one is found or the search is terminated. Previously, we described a policy that uses NP for disclosing edge balance information in a manner that provides privacy. Here, we consider how this information may be used to improve the effectiveness of path planning.

For each $e \in E$, let $b^e \in [0, c^e]$ denote the corresponding current (unknown) balance value, and let s_n^e denote the corresponding edge balance subsequence disclosed by the policy defined in Algorithm 1. Let $b_p^e \in [0, c^e]$ denote the predicted value of b^e based on s_n^e. This prediction could be performed using a forecasting model such as ARIMA (AutoRegressive Integrated Moving Average). Due to forecasting errors, b_p^e may not equal b^e.

Path planning involves searching the space of payment paths to find a low-fee, feasible payment path for making a given payment. This task is performed with respect to the predicted balance values described above. Therefore, inaccurately predicted balance values increase the difficulty of path planning. To demonstrate this, we consider a policy where an edge $e \in E$ is considered to be feasible with respect to forwarding a payment of amount a if and only if the condition $a \leq b_p^e$ is satisfied.

Let us first consider the problem of determining if a payment of amount a can be forwarded using a payment path containing a single edge $e \in E$. If $b^e < a \leq b_p^e$, a false positive will occur whereby it is incorrectly determined that e can successfully forward the payment in question. On the other hand, if $b_p^e < a \leq b^e$, a false negative will occur whereby it may be incorrectly determined that e cannot successfully forward the payment in question. Note that, if the predicted balance b_p^e equals the current balance b^e, neither of the conditions $b^e < a \leq b_p^e$ or $b_p^e < a \leq b^e$ will be satisfied for any values of a. Hence, a false positive or false negative respectively cannot occur.

Let $b_p^e = b^e + m$ where $m \in [0, c^e]$ is an error term. A false positive occurs if the payment amount a is in the interval $(b^e, b_p^e]$. The length of this interval is $b_p^e - b^e = m$. Assuming payment amounts are uniformly distributed on the interval $[0, c^e]$, the probability of a false positive is the length of this interval divided by the total length of the support.

$$Pr[\textsc{False Positive}|b_p^e] = \frac{m}{c^e} \tag{15}$$

On the other hand, let $b_p^e = b^e - m$ where $m \in [0, c^e]$. Assuming payment amounts are uniformly distributed on the interval $[0, c^e]$, the probability of a false negative is defined as follows.

$$Pr[\textsc{False Negative}|b_p^e] = \frac{m}{c^e} \tag{16}$$

The above analysis demonstrates that the probabilities of both a false positive and a false negative are monotonically increasing functions of m that converge to 0 at m equal to 0.

Next consider the problem of determining if a payment of amount a can be forwarded using a payment path $e_1, e_2, \ldots, e_n$ containing n edges. To simplify our analysis, we assume that each edge in this path has the same capacity c^e and charge zero fees for forwarding payments. We adopt the most realistic model where the prediction error is unique for each edge, such that $b_p^{e_i} = b^{e_i} + m^i$, where each error term $m^i \in [-c^e, c^e]$ can be positive, negative, or zero. A false positive will occur if the following condition is satisfied.

$$(\forall i, a \leq b_p^{e_i}) \wedge (\exists j, a > b^{e_j}) \tag{17}$$

The first term in this condition models that the payment is predicted feasible while the second term models that the payment is actually not feasible. The first term implies that $a \leq \min_{i=1\ldots n}(b_p^{e_i})$ while the second term implies that $a > \min_{i=1\ldots n}(b^{e_i})$. Combining these, a false positive occurs if the payment amount a falls within the following range.

$$\min_{i=1\ldots n}(b^{e_i}) < a \leq \min_{i=1\ldots n}(b_p^{e_i}) \tag{18}$$

The length of the interval $(\min_{i=1\ldots n}(b^{e_i}), \min_{i=1\ldots n}(b_p^{e_i})]$ equals $\min_{i=1\ldots n}(b_p^{e_i}) - \min_{i=1\ldots n}(b^{e_i})$ if $\min_{i=1\ldots n}(b_p^{e_i}) > \min_{i=1\ldots n}(b^{e_i})$ and 0 otherwise. In the latter case, a false positive is impossible. Assuming that payment amounts follow a uniform distribution on the interval $[0, c^e]$, the probability of a path-level false positive is defined as follows.

$$\begin{aligned} Pr[\text{False Positive}|\{b_p^{e_i}\}] &= \frac{\max(0, \min_{i=1\ldots n}(b_p^{e_i}) - \min_{i=1\ldots n}(b^{e_i}))}{c^e} \\ &= \frac{\max(0, \min_{i=1\ldots n}(b^{e_i} + m^i) - \min_{i=1\ldots n}(b^{e_i}))}{c^e} \end{aligned} \tag{19}$$

A path-level false negative occurs if the path is predicted to fail but would have succeeded. This happens if a is in the range $\min_i(b_p^{e_i}) < a \leq \min_i(b^{e_i})$. Assuming that payment amounts follow a uniform distribution on the interval $[0, c^e]$, the probability of a path-level false negative is defined as follows.

$$\begin{aligned} Pr[\text{False Negative}|\{b_p^{e_i}\}] &= \frac{\max(0, \min_{i=1\ldots n}(b^{e_i}) - \min_{i=1\ldots n}(b_p^{e_i}))}{c^e} \\ &= \frac{\max(0, \min_{i=1\ldots n}(b^{e_i}) - \min_{i=1\ldots n}(b^{e_i} + m^i))}{c^e} \end{aligned} \tag{20}$$

With a mix of positive and negative errors, the type of error a path is vulnerable to is not fixed. It's an emergent property determined by the specific pattern of errors across all edges, which dictates whether the predicted minimum balance is higher or lower than the actual minimum. To demonstrate this consider a payment path e_1, e_2 containing two edges where $(b^{e_1}, b^{e_2}) = (100, 110)$. If $(m^1, m^2) = (+20, -5)$, the predicted balances will be $(b_p^{e_1}, b_p^{e_2}) = (100 + 20, 110 - 5) = (120, 105)$. Since $\min_{i=1\ldots 2}(b_p^{e_i}) > \min_{i=1\ldots 2}(b^{e_i})$, the path is now vulnerable to false positives for any payment a in the range $(100, 105]$. As a second example, consider a payment path e_1, e_2 containing two edges where $(b^{e_1}, b^{e_2}) = (100, 110)$. If $(m^1, m^2) = (-5, -20)$, the predicted balances will be $(b_p^{e_1}, b_p^{e_2}) = (100 - 5, 110 - 20) = (95, 90)$. Since $\min_{i=1\ldots 2}(b_p^{e_i}) < \min_{i=1\ldots 2}(b^{e_i})$, the path is now vulnerable to false negatives for any payment a in the range $(90, 100]$.

This generalised analysis reveals that while single-edge error probabilities are simple functions, path-level probabilities are complex and depend on the entire state of the path. However, the core conclusion remains: improving prediction accuracy by minimising the magnitude of all individual errors ($|m^i| \longrightarrow 0$) ensures that $\min(b_p^{e_i}) \approx \min(b^{e_i})$. This symmetrically reduces the conditions that allow for both types of path-level errors, highlighting the usefulness of accurate balance forecasts for effective path planning.

4 EXPERIMENTAL RESULTS & ANALYSIS

In this section, we evaluate the proposed method for disclosing edge balance information through both theoretical and experimental analysis. The private nature of the LN means no public dataset of historical payments exists, making simulation the only viable approach for empirical evaluation. Therefore, our experimental analysis uses a simulation of synthetic payments on a real snapshot of the LN. The section is structured as follows. First, in Section 4.1, we describe the data used for both the theoretical and experimental analyses. Next, in Section 4.2, we present the theoretical analysis of the privacy parameters. Following this, in Section 4.3, we detail the methodology for our path planning simulation. Finally, in Section 4.4, we present and discuss the simulation's findings.

4.1 Data

We saved a snapshot of the LN on 13 October 2023 using the Lightning Network Daemon (lnd) peer implementation[1]. From this snapshot, we first removed all channels that did not have a fee policy, i.e. all channels that cannot be used in a payment path. The snapshot contains a large proportion of isolated peers that are not contained in any public channels. Such peers may be contained in unannounced private channels. However, we do not have knowledge of these channels and, therefore, we removed all isolated peers. Consequently, the LN snapshot was reduced to 12,952 peers and 55,912 channels. The corresponding graph representation $G = (V, E)$ contained 12,952 vertices and 111,824 edges (each channel corresponds to two edges). On the date the above snapshot was obtained, 1 Satoshi (the atomic unit of Bitcoin currency used by the LN) had an approximate value of 0.0002 Great British Pounds (GBP). The mean and median LN snapshot channel capacities were 6,584,164 Satoshis and 2,000,000 Satoshis. The 10th and 90th percentiles of LN snapshot channel capacities were 150,000 Satoshis and 11,000,000 Satoshis respectively.

A significant part of our analysis concerns simulating a sequence of payment attempts on the above LN snapshot and counting the number of successful attempts. As discussed previously, the LN peers do not disclose payment information. Therefore, there exists no publicly available dataset of historical LN payments that could be used to inform simulations. To overcome this challenge we used the following approach to generate a synthetic sequence of payments that we hope is a close approximation to a potentially real set of payments attempts. However, there currently exists no way to verify this.

A payment is defined by two parameters: (1) a payment source and destination pair, and (2) a payment value. To generate a sequence of synthetic payments, we made the following assumptions regarding the values of these parameters. An LN peer that forwards payments will know the time and value of these payments but not the source or destination, due to the use of onion routing. The company River aggregated data from several companies that operate LN peers and forward a large number of payments. They published summary statistics about the payment values from January to August in 2023 [21]. In August 2023 the average payment value was 44,700 Satoshis. However, a visual inspection of the bar charts in the article suggests this value was significantly higher in several of the previous months in 2023. Furthermore, for all months considered, the majority of payment values were less than 1,000,000 Satoshis. Given this information, we assume that payment values are sampled uniformly at random from the interval $[1, 1000000]$. We argue that it is a reasonable assumption that attempted payments are a function of the LN graph topology. A LN channel will not be created between a random pair of peers and will not have a random capacity. Instead, we argue that an LN channel is generally created to support a large number of intended future payments. Given this, we assume that payments are made between random pairs of source and destination vertices where there exists a corresponding feasible path in the LN graph

[1] https://docs.lightning.engineering/

given that edges have a balance equal to one-tenth of the channel capacity. Finally, we assume that payments are not independent but instead similar payments are frequently repeated.

Given the above assumptions, we used the following two-step approach to generate a sequence of payments $P = (p_1, p_2, \ldots, p_m)$ containing m elements. In the first step, we generated a set of payments $Q = \{q_1, q_2, \ldots, q_{m'}\}$ containing m' payments where $m' \leq m$. In the second step, we sampled with replacement a sequence of m payments from this set. The initial set of payments Q was generated using rejection sampling as follows. First, we specified the balance of each edge in the LN snapshot to equal one-tenth of the corresponding channel capacity. Next, we sampled the source and destination vertices by sampling uniformly at random a pair of distinct vertices and sampled the payment amount by sampling uniformly at random an integer in the interval $[1, 1000000]$. Finally, we determined whether the payment in question was feasible. If the payment was feasible, we added it to the set Q; otherwise, we did not add it to the set. We repeated this process of sampling feasible payments until the set contained m' elements. In our analysis, the parameters m and m' were set to $25,000$ and $1,000$ respectively. Therefore, the expected number of times each element in the set of payments Q appeared in the sequence of payments P was 25. The mean and standard deviation of the payment values in the sequence of payments was $175,188$ and $202,519$ respectively.

4.2 Analysis of Privacy Parameters

In Algorithm 1, an edge discloses its balance only when the privacy parameters ε and δ, defined in Theorem 2, fall below specified security thresholds. The number of payments n required to meet this condition is driven by the interplay between the channel's properties and the payment statistics. This required number of payments also depends critically on the adversary's prior knowledge. This prior knowledge is measured by γ, which represents the fraction of payment instances in the set of n that the adversary is assumed to know. To simplify the following analysis, we assume that the individual payment values are independent and identically distributed (i.i.d.). This implies they share a common variance σ^2 and a common third absolute central moment $\rho^3 = \mathbb{E}(|X - \mu|^3)$.

In the following sections, we first analyse the baseline case ($\gamma = 0$), extend this result to the general case ($\gamma > 0$), and finally, ground these theoretical findings with a quantitative numerical analysis.

4.2.1 Baseline Case ($\gamma = 0$).

We begin with the case where an adversary has no prior knowledge of any payments ($\gamma = 0$). In this scenario, the summations in the theorem are over all n payments.

First, we analyse the behaviour ε. We derive its specific relationship with n from the general formula by setting $\gamma = 0$ and applying the i.i.d. assumption.

$$\varepsilon = \sqrt{\frac{\Delta^2 \ln(n)}{\sum_{i=1}^{n} \mathrm{Var}(X_i)}} = \sqrt{\frac{\Delta^2 \ln(n)}{n\sigma^2}} = \frac{\Delta}{\sigma}\sqrt{\frac{\ln(n)}{n}} \tag{21}$$

This result shows ε is directly proportional to the ratio of the channel capacity to the payment standard deviation ($\frac{\Delta}{\sigma}$). This ratio is typically very large for a fundamental reason: the channel capacity, which defines the sensitivity Δ, will in many cases be significantly larger than the value of any individual payment. Since Δ is much larger than the payment values themselves, it is even more significantly larger than their standard deviation σ. Consequently, a correspondingly large n is required to counteract this high ratio and make ε acceptably small.

Next, we analyse the behaviour of δ. We start with the general formula from the theorem and set $\gamma = 0$.

$$\delta = \frac{1.12 \sum_{i=1}^{n} \mathbb{E}(|X_i - \mu_i|^3)}{\left(\sum_{i=1}^{n} \mathrm{Var}(X_i)\right)^{\frac{3}{2}}} (1 + e^{\varepsilon}) + \frac{4}{5\sqrt{n}} \tag{22}$$

To determine the relationship with n, we simplify the first term. Applying the i.i.d. assumption, the sums in the numerator and denominator become

$$\sum_{i=1}^{n} \mathbb{E}(|X_i - \mu|^3) = n\rho^3. \tag{23}$$

$$\left(\sum_{i=1}^{n} \mathrm{Var}(X_i)\right)^{\frac{3}{2}} = \left(n\sigma^2\right)^{\frac{3}{2}} = n^{\frac{3}{2}}\sigma^3. \tag{24}$$

Substituting these back into the fraction and simplifying the powers of n demonstrates the proportionality.

$$\frac{1.12 \cdot n\rho^3}{n^{3/2}\sigma^3} (1 + e^{\varepsilon}) = \frac{1.12\rho^3}{\sigma^3} \cdot \frac{1}{\sqrt{n}} (1 + e^{\varepsilon}) \tag{25}$$

Since the first term simplifies to a form proportional to $1/\sqrt{n}$, and the second term is also proportional to $1/\sqrt{n}$, the entire expression for δ reliably converges to zero as n increases.

4.2.2 General Case ($\gamma > 0$).

To generalise this analysis, we reframe the problem: the case where an adversary knows a fraction γ of n payments can be treated as an equivalent baseline ($\gamma = 0$) analysis on a smaller, effective dataset of $n' = (1 - \gamma)n$ unknown payments. We can therefore derive the general formulas by taking the baseline results and substituting n with this effective size n'.

First, we analyse ε. Applying the substitution $n \longrightarrow n'$ to the baseline result for ε

$$\varepsilon = \frac{\Delta}{\sigma}\sqrt{\frac{\ln(n')}{n'}} = \frac{\Delta}{\sigma}\sqrt{\frac{\ln((1-\gamma)n)}{(1-\gamma)n}} \tag{26}$$

Next, we analyse the behaviour of δ. Applying the substitution $n \longrightarrow n'$ to the derived components of δ

$$\delta = \left(\frac{1.12\rho^3}{\sigma^3} \cdot \frac{1}{\sqrt{n'}} (1 + e^{\varepsilon})\right) + \frac{4}{5\sqrt{n'}} = \frac{1}{\sqrt{(1-\gamma)n}} \left(\frac{1.12\rho^3}{\sigma^3}(1 + e^{\varepsilon}) + \frac{4}{5}\right) \tag{27}$$

This generalization shows that as adversary knowledge γ increases, the effective number of payments $(1 - \gamma)n$ decreases, causing both ε and δ to become larger and weaken the privacy guarantee.

4.2.3 Numerical Analysis.

The theoretical analysis reveals how the privacy parameters ε and δ depend on the number of aggregated payments n, the channel sensitivity Δ, the payment statistics (σ, ρ^3), and adversary knowledge γ. To ground these abstract relationships, we now provide a quantitative analysis for the baseline case ($\gamma = 0$) using realistic network parameters.

A key challenge is selecting appropriate thresholds for the privacy parameters. As highlighted by Dwork et al. [7], there is no clear consensus on the best values for ε and δ in DP, a fact that also applies to NP. However, a recent article by the National Institute of Standards and Technology (NIST) makes some suggestions for the value of the ε parameter based on their experience. They state that a value in the interval $[0, 5]$ is conservative, a value in the interval $[5, 20]$ also provides robust privacy protection in a variety of settings, and a value greater than 20 may still provide meaningful privacy protection [15].

Capacity / n	2	4	8	16	32	64	128
Mean	6.7, 0.8	6.7, 0.4	5.8, 0.1	4.7, 0.0	3.8, 0.0	2.9, 0.0	2.2, 0.0
Median	2.0, 0.0	2.0, 0.0	1.8, 0.0	1.4, 0.0	1.1, 0.0	0.9, 0.0	0.7, 0.0
10th percentile	1.0, 0.0	1.0, 0.0	0.9, 0.0	0.7, 0.0	0.6, 0.0	0.4, 0.0	0.3, 0.0
90th percentile	11.2, 30.5	11.2, 25.1	9.7, 3.5	7.9, 0.5	6.3, 0.1	4.9, 0.0	3.7, 0.0

Table 1. Estimates of the expected values of ε and δ for edges with different edge capacities and different values of n are displayed. The mean, median, 10th percentile and 90th percentile edge capacities are $6, 584, 164$, $2, 000, 000$, $150, 000$ and $11, 000, 000$ respectively.

For this analysis, we used the four representative channel capacities (Mean, Median, 10th, and 90th percentile) from Section 4.1 as our values for sensitivity, Δ. To model bi-directional payments, consistent with the theoretical framework, we sampled payment values from a uniform distribution on a symmetric interval $[-\beta, \beta]$. A positive value represents a payment forwarded by the edge (decreasing its balance), while a negative value represents a payment in the opposite direction through the channel (increasing its balance). Based on empirical data suggesting most real-world payments are below 1,000,000 Satoshis, we set β to be the minimum of 1,000,000 and the channel capacity in question. For each scenario, we estimated the expected values of ε and δ by averaging the results of 2,000 simulations.

The results, displayed in Table 1, align with the theoretical derivation in the baseline case. As predicted by the formula in Equation 21, for any given n, the value of ε is consistently highest for the 90th percentile capacity, which has the largest sensitivity Δ. This quantitatively confirms that channels with larger capacities require a greater number of payments to achieve the same privacy level. Adopting a conservative threshold of $\varepsilon < 5$ and $\delta < 5$ based on the NIST guidance, the table shows that for channels with the mean capacity or less, this is achieved for n at or below 16. For the larger 90th percentile channels, a more substantial aggregation of n at or below 64 payments is required to meet the same threshold.

In conclusion, this empirical analysis provides quantitative validation for the theoretical framework, demonstrating that channels with higher capacities (larger Δ) require a significantly larger number of payments n to be aggregated. However, this does not necessarily mean they will disclose their balances less frequently over time. High-capacity channels often serve as major routing hubs and are likely to have a much higher payment throughput. The temporal frequency of disclosure depends on the ratio of the required aggregation count (n) to the payment rate. It is plausible that a high-volume channel could achieve its larger aggregation target in the same amount of time, or even faster, than a low-volume channel. Therefore, the practical implication for Algorithm 1 is that while high-capacity channels must aggregate more payments per disclosure, their disclosure frequency in terms of time is a dynamic property of their payment volume.

4.3 Path Planning Simulation: Methodology

In this section we describe the methodology used to simulate the sequence of payments $P = (p_1, p_2, \ldots, p_m)$ described in Section 4.1. When simulating this sequence of payments, we assumed that initially the capacity of each channel was randomly distributed between the balances of the corresponding pair of edges. The payments in P were attempted sequentially in the order they appear in the sequence using the following approach.

First, a predictive model is used to determine b_p^e, which denotes the predicted value of the corresponding true balance b^e, for each edge $e \in E$. Our simulation uses a simple predictive model that assigns b_p^e to equal the most recently disclosed balance for edge e. Next, a conservative balance estimate $b_p^{e\prime} = b_p^e \times \alpha$, where $\alpha \in [0, 1]$, is calculated for each edge. This conservative approach is motivated by the fact that payments are atomic; if any single edge has insufficient funds, the entire payment fails. Next, a search for a lowest-fee payment path is performed using a variant of Dijkstra's algorithm [4]. This variant incorporates the balance constraint directly into its search process: it explores paths using the channel fee policies as edge weights but will only consider traversing an edge if its conservatively estimated balance $b_p^{e\prime}$ is sufficient for the payment amount plus any necessary fees.

If the above search finds a payment path, it is then validated against the true balances: the payment is a success if every edge in the path has a sufficient true balance. In this case, the true balances of the edges along the path are updated. For each of these edges, the new payment value is appended to its internal list of aggregated payments, x, as shown in Algorithm 1. The algorithm then re-evaluates the privacy conditions based on the new list. If an edge's update condition is met, its publicly disclosed balance is updated to the new true balance. If the path fails the true balance check, or if no path was found, the payment is a failure, and the state of the network remains unchanged.

We use two quantitative metrics to evaluate the performance of the disclosure policy. The total number of successful payments directly measures the network's utility; a higher count indicates that the disclosed balance information, though intentionally updated infrequently, remains accurate enough for effective path planning. The mean balance estimation error, measured as the absolute difference between the true and predicted values, directly quantifies the accuracy of this information. A lower error indicates a more accurate prediction, while a higher error measures the performance cost of increased privacy, where less frequent updates may cause the predicted balance to deviate from the true balance.

4.4 Path Planning Simulation: Results and Discussion

In this section, we present the results of the path planning simulation, which provides quantitative support for the theoretical argument from Section 3.3.4 - that accurate edge balance predictions support effective path planning. The analysis is structured as follows: first, we establish the performance of the upper and lower bound baselines. Next, we investigate the core trade-off between privacy and utility by adjusting the privacy parameters of the proposed disclosure method. Finally, we examine the impact of partial network adoption on path planning performance.

4.4.1 Baseline Performance.

To contextualise the performance of the proposed method, we compare it against two baselines. The first baseline models a scenario with real-time balance information where balance predictions have zero error. This baseline represents a performance upper bound. The second baseline models a scenario with no disclosed balance information, where balances are predicted using a 50% capacity heuristic, which assumes each balance is equal to half the channel capacity. This baseline represents a performance lower bound. The performance of the proposed disclosure method is expected to fall between these two extremes. Table 2 displays the number of successful payments and the mean balance estimation error after all payment attempts as a function of α for each baseline. Recall that the total number of payment attempts is 25,000. For a given edge, we measure the error of the corresponding balance estimate as the absolute difference between the true and predicted values.

Examining this table reveals several key dynamics. The perfect estimation baseline significantly outperforms the 50% capacity baseline in successful payments across all values of α. The performance of the perfect baseline is optimal at $\alpha = 1.0$ with 19,072 successes (76%) and degrades as α decreases. This is expected; with perfect information, any added conservatism ($\alpha < 1.0$) is counterproductive, as it causes the path planning algorithm to discard feasible payment paths.

Baseline / α	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
50% Capacity	3,845	3,921	3,916	3,911	3,890	3,873	3,882	4,052	4,173	4,491
estimation	1,648k	1,648k	1,648k	1,648k	1,648k	1,648k	1,648k	1,649k	1,650k	1,651k
Perfect	19,072	18,980	18,845	18,698	18,493	18,214	17,774	17,040	15,647	11,906
estimation	0	0	0	0	0	0	0	0	0	0

Table 2. For the 50% capacity heuristic and perfect estimation baselines the number of successful payments followed by the mean balance estimation error after all payment attempts as a function of α is displayed. The symbol k equals 10^3.

Conversely, the 50% capacity baseline's performance, while non-monotonic, is generally better at more conservative α values, achieving its maximum of 4,491 successes at $\alpha = 0.1$. This occurs because the 50% capacity estimate is poor, and a high degree of conservatism forces the algorithm to select only those paths that appear massively over-provisioned, which have a higher chance of being feasible. As expected, the mean balance estimation error is consistently large for the 50% capacity baseline (around 1,650,000 Satoshis), while it is zero for the perfect estimation baseline.

4.4.2 The Privacy vs. Utility Trade-off.

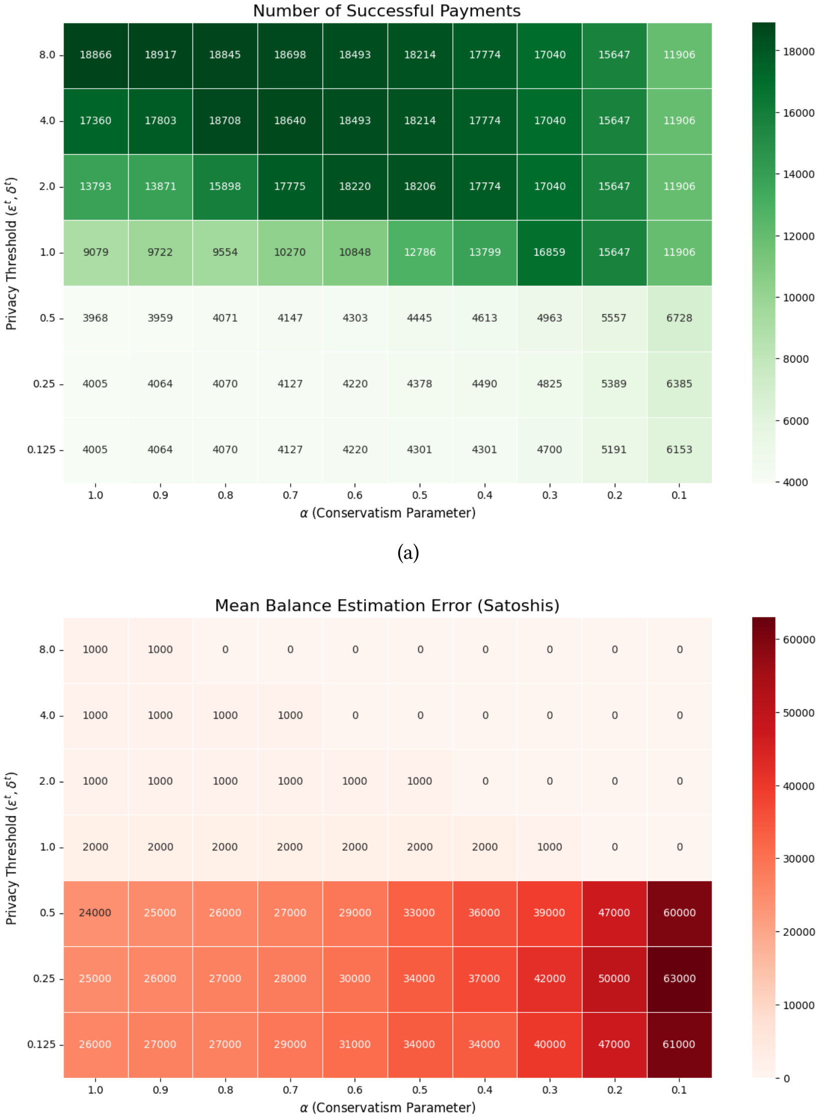
To investigate the relationship between privacy and path planning support, we simulated a scenario where each edge implements Algorithm 1 using a range of privacy thresholds ε^t and δ^t where $\varepsilon^t = \delta^t$. Figure 4 visualises the results of this simulation.

The heatmaps illustrate a clear trade-off. In Figure 4(a), for high thresholds (e.g., $\varepsilon^t = \delta^t = 8.0$), the number of successful payments is high, approaching the performance of the perfect estimation baseline. Conversely, as the privacy thresholds become stricter (e.g., $\varepsilon^t = \delta^t \leq 0.5$), the success rate declines sharply. This corresponds directly with the mean balance estimation error, shown in Figure 4(b). Stricter privacy requires a larger number of payments n to be aggregated between disclosures, causing the disclosed balance information to become less accurate and leading to a significant increase in estimation error. While the strictest settings can underperform the 50% capacity baseline's best result of 4,491 successes, a moderately strict setting (e.g., $\varepsilon^t = \delta^t = 1.0$) can still yield over 10,000 successes, far exceeding this approach.

4.4.3 Impact of Partial Network Adoption.

In the final experiment, we consider the case where only a fraction $\tau \in [0, 1]$ of channels participate in the proposed balance disclosure method. For these participating channels, we assume a moderately strict privacy setting where $\varepsilon^t = \delta^t = 1.0$. The balances for all non-participating channels are estimated using the 50% capacity heuristic. Figure 5 visualizes the number of successful payments and the mean balance estimation error as a function of the participation rate τ and the path planning conservatism parameter α.

The results demonstrate a graceful degradation in performance as participation decreases. As shown in Figure 5(a), the number of successful payments declines as τ decreases, but even with only 50% participation ($\tau = 0.5$), the system achieves over 6,300 successes, significantly outperforming the 4,491 successes of the pure 50% capacity heuristic. As expected, as τ approaches zero, the performance converges to that of the heuristic baseline. Figure 5(b) shows that the mean estimation error generally increases as fewer channels participate, which is the expected outcome of relying more heavily on the high-error 50% capacity heuristic. However, it also reveals that for $\tau > 0$, the error paradoxically decreases at the most conservative α value of 0.1. This is likely because extreme conservatism forces the path planning algorithm to select only paths it is highly confident

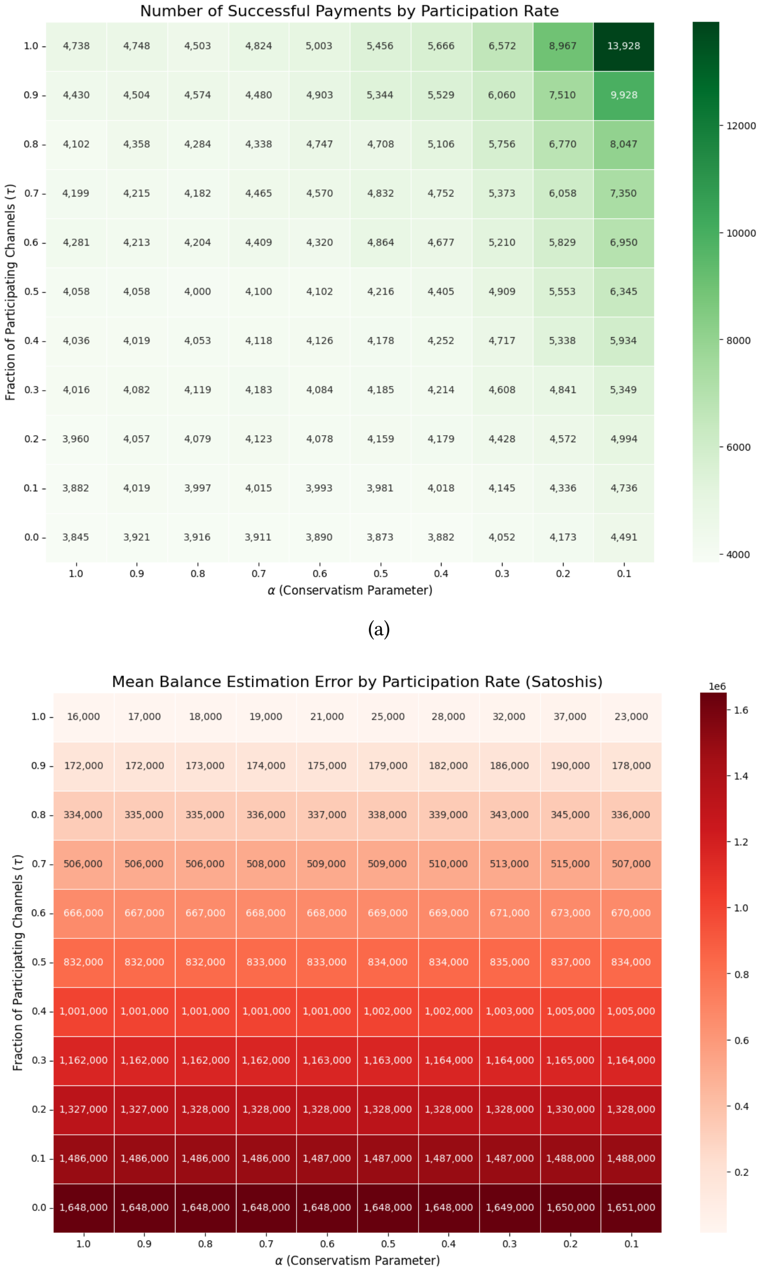


(a)

(b)

Fig. 4. Heatmaps illustrating the trade-off between privacy and path planning performance. (a) The number of successful payments out of 25,000 attempts. Higher values (greener cells) indicate better path planning performance. (b) The mean balance estimation error in Satoshis. Lower values (greener cells) indicate more accurate balance information. Both heatmaps plot the common privacy threshold ($\varepsilon^t = \delta^t$) against the path planning conservatism parameter (α).

in, skewing the sample of successful payments towards those with more accurate (and often more recent) balance information.



(a)

(b)

Fig. 5. Heatmaps illustrating the impact of partial network adoption of the privacy-preserving disclosure policy. (a) The number of successful payments out of 25,000 attempts. Performance degrades gracefully as the fraction of participating channels (τ) decreases. (b) The mean balance estimation error in Satoshis. The error increases as fewer channels participate, increasing reliance on the high-error heuristic. Both heatmaps plot the participation rate (τ) against the path planning conservatism parameter (α).

5 CONCLUSIONS

This work addresses the fundamental conflict between payment privacy and path planning efficiency in Payment Channel Networks (PCNs). We have proposed and evaluated a novel, decentralised method for disclosing channel balance information that leverages the principles of Noiseless Privacy (NP). By disclosing balance states only after a dynamically determined number of payments have occurred, our method provides quantifiable privacy guarantees without adding noise to the data, thereby preserving its utility for path planning. This approach avoids the prohibitive communication overhead of real-time updates and circumvents the inapplicability of traditional Differential Privacy (DP) in this context. Given the novel nature of this work, we conclude by outlining several possible directions for future research.

The most direct extension is to improve the prediction accuracy of the path planning model. In our simulation, we used a simple model that assumes the current true balance is equal to the most recently disclosed balance. Future work could explore more sophisticated predictive models that might, for instance, use the complete time series of balance disclosures to produce a more accurate estimate of the true balance, potentially improving path planning success without weakening the core privacy guarantee.

Beyond improving the predictive model, the privacy policy itself could be made more adaptive. The current method uses fixed thresholds, ε^t and δ^t, for all disclosures. A further avenue for research is the development of dynamic policies where a vertex could adjust its privacy level based on network conditions or economic incentives. For instance, an edge with high liquidity seeking to attract more payments could temporarily relax its privacy settings. Investigating the game-theoretic implications of such adaptive policies, where vertices compete on both privacy and utility, represents a compelling direction for future work.

While these refinements would improve the utility of the method, its privacy guarantees could also be extended. The proposed method provides privacy with respect to individual payments. However, in some situations, it may be necessary to provide privacy with respect to larger sets or flows of payments. This is an issue of privacy granularity that is concerned with providing different scales or levels of privacy [7]. Using NP methods to provide such multi-level privacy represents another opportunity for future research.

In addition to extending the privacy model, it is also crucial to analyse its interaction with other LN privacy technologies, such as blinded paths. It is unclear if providing even infrequent balance information creates new attack vectors or synergies when combined with recipient anonymity. For example, an adversary with partial balance knowledge might be able to more easily de-anonymise a blinded path by correlating balance changes with the limited set of possible routes. Understanding and mitigating these potential second-order effects is crucial for developing a holistic privacy framework for the network.

Finally, looking beyond the immediate application of path planning, the long-term statistical data generated by this method opens a new research direction in higher-level network optimisation. The infrequent but privacy-preserving balance updates from across the network could be aggregated over time to identify liquidity trends, under-utilised channels, and network bottlenecks. Future research could explore how this aggregated data could inform automated, decentralised decisions about where to allocate new capital or perform rebalancing operations, thereby improving the overall efficiency and reliability of the network.

REFERENCES

[1] Andreas M Antonopoulos, Olaoluwa Osuntokun, and René Pickhardt. 2021. *Mastering the Lightning Network.* O'Reilly Media.

[2] Ferenc Béres, István András Seres, and András A Benczúr. 2021. A cryptoeconomic traffic analysis of Bitcoin's Lightning Network. (2021).

[3] Raghav Bhaskar, Abhishek Bhowmick, Vipul Goyal, Srivatsan Laxman, and Abhradeep Thakurta. 2011. Noiseless database privacy. In *International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea.* 215–232.

[4] Padraig Corcoran and Rhyd Lewis. 2025. An analysis of the correctness and computational complexity of path planning in Payment Channel Networks. *The Journal of Financial Technology* (2025).

[5] Maya Dotan, Saar Tochner, Aviv Zohar, and Yossi Gilad. 2022. Twilight: A differentially private payment channel network. In *USENIX Security Symposium.* 555–570.

[6] Yitao Duan. 2009. Differential privacy for sum queries without external noise. In *ACM Conference on Information and Knowledge Management.*

[7] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. 2019. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality* 9, 2 (2019).

[8] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[9] Krzysztof Grining and Marek Klonowski. 2017. Towards extending noiseless privacy: Dependent data and more practical approach. In *ACM on Asia Conference on Computer and Communications Security.* 546–560.

[10] Abdelatif Hafid, Abdelhakim Senhaji Hafid, and Mustapha Samih. 2020. Scaling blockchains: A comprehensive survey. *IEEE Access* 8 (2020), 125244–125262.

[11] Jordi Herrera-Joancomartí, Guillermo Navarro-Arribas, Alejandro Ranchal-Pedrosa, Cristina Pérez-Solà, and Joaquin Garcia-Alfaro. 2019. On the difficulty of hiding the balance of lightning network channels. In *ACM Asia Conference on Computer and Communications Security.* 602–612.

[12] George Kappos, Haaroon Yousaf, Ania Piotrowska, Sanket Kanjalkar, Sergi Delgado-Segura, Andrew Miller, and Sarah Meiklejohn. 2021. An empirical analysis of privacy in the lightning network. In *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part I 25.* Springer, 167–186.

[13] Arad Kotzer and Ori Rottenstreich. 2023. Braess Paradox in Layer-2 Blockchain Payment Networks. In *IEEE International Conference on Blockchain and Cryptocurrency.* 1–9.

[14] Satwik Prabhu Kumble, Dick Epema, and Stefanie Roos. 2021. How lightning's routing diminishes its anonymity. In *Proceedings of the 16th International Conference on Availability, Reliability and Security.* 1–10.

[15] Joseph Near and David Darais. 2022. Differential privacy: Future work & open challenges. *Cybersecurity insights* (2022). https://www.nist.gov/blogs/cybersecurity-insights/differential-privacy\protect\penalty\z@-future-work-open-challenges [Accessed: 26 Oct. 2023].

[16] Utz Nisslmueller, Klaus-Tycho Foerster, Stefan Schmid, and Christian Decker. 2020. Toward active and passive confidentiality attacks on cryptocurrency off-chain networks. *arXiv preprint arXiv:2003.00003* (2020).

[17] Rene Pickhardt. 2022. bolt14: Sharing X bit of liquidity information in the friend of a friend network of sender and receiver by Rene Pickhardt · pull request 780 · Lightning/bolts. https://github.com/lightning/bolts/pull/780 [Accessed: 26 Oct. 2023].

[18] Rene Pickhardt, Sergei Tikhomirov, Alex Biryukov, and Mariusz Nowostawski. 2021. Security and privacy of lightning network payments with uncertain channel balances. *arXiv preprint arXiv:2103.08576* (2021).

[19] Joseph Poon and Thaddeus Dryja. 2016. The bitcoin lightning network: Scalable off-chain instant payments.

[20] Sonbol Rahimpour and Majid Khabbazian. 2022. Torrent: Strong, fast balance discovery in the Lightning Network. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC).* IEEE, 1–7.

[21] River. 2023. *Research Report: The Lightning Network Grew by 1212% in 2 Years, Why It's important to Pay Attention.* Technical Report. River.

[22] Elias Rohrer and Florian Tschorsch. 2020. Counting down thunder: Timing attacks on privacy in payment channel networks. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies.* 214–227.

[23] Vikash Singh, Matthew Khanzadeh, Vincent Davis, Harrison Rush, Emanuele Rossi, Jesse Shrader, and Pietro Lio'. 2025. Bayesian Binary Search. *Algorithms* 18, 8 (2025), 452.

[24] Vibhaalakshmi Sivaraman, Shaileshh Bojja Venkatakrishnan, Kathleen Ruan, Parimarjan Negi, Lei Yang, Radhika Mittal, Giulia Fanti, and Mohammad Alizadeh. 2020. High throughput cryptocurrency routing in payment channel networks. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20).* 777–796.

[25] Weizhao Tang, Weina Wang, Giulia Fanti, and Sewoong Oh. 2020. Privacy-utility tradeoffs in routing cryptocurrency over payment channel networks. *ACM on Measurement and Analysis of Computing Systems* 4, 2 (2020), 1–39.

[26] Bastien Teinturier. 2023. Route Blinding. https://github.com/lightning/bolts/blob/master/\protect\penalty\z@proposals/route-blinding.md [Accessed: 26 Oct. 2023].

[27] Sergei Tikhomirov, Rene Pickhardt, Alex Biryukov, and Mariusz Nowostawski. 2020. Probing channel balances in the lightning network. *arXiv preprint arXiv:2004.00333* (2020).

[28] Saar Tochner and Stefan Schmid. 2020. On search friction of route discovery in offchain networks. In *IEEE International Conference on Blockchain*. 257–264.

[29] Gijs van Dam and Rabiah Abdul Kadir. 2022. Hiding payments in lightning network with approximate differentially private payment channels. *Computers & Security* 115 (2022), 102623.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009