Privacy-Utility Tradeoffs in Routing Cryptocurrency over Payment Channel Networks

WEIZHAO TANG, Carnegie Mellon University, USA
WEINA WANG, Carnegie Mellon University, USA
GIULIA FANTI, Carnegie Mellon University, USA
SEWOONG OH, University of Washington, USA

Payment channel networks (PCNs) are viewed as one of the most promising scalability solutions for cryptocurrencies today [28]. Roughly, PCNs are networks where each node represents a user and each directed, weighted edge represents funds escrowed on a blockchain; these funds can be transacted only between the endpoints of the edge. Users efficiently transmit funds from node A to B by relaying them over a path connecting A to B, as long as each edge in the path contains enough balance (escrowed funds) to support the transaction. Whenever a transaction succeeds, the edge weights are updated accordingly. In deployed PCNs, channel balances (i.e., edge weights) are not revealed to users for privacy reasons; users know only the initial weights at time 0. Hence, when routing transactions, users typically first guess a path, then check if it supports the transaction. This guess-and-check process dramatically reduces the success rate of transactions. At the other extreme, knowing full channel balances can give substantial improvements in transaction success rate at the expense of privacy. In this work, we ask whether a network can reveal *noisy* channel balances to trade off privacy for utility. We show fundamental limits on such a tradeoff, and propose noise mechanisms that achieve the fundamental limit for a general class of graph topologies. Our results suggest that in practice, PCNs should operate either in the low-privacy or low-utility regime; it is not possible to get large gains in utility by giving up a little privacy, or large gains in privacy by sacrificing a little utility.

CCS Concepts: • **Security and privacy → Privacy-preserving protocols**; • **Networks → Network privacy and anonymity**; *Network algorithms.*

Additional Key Words and Phrases: Blockchain, privacy, p2p network

ACM Reference Format:
Weizhao Tang, Weina Wang, Giulia Fanti, and Sewoong Oh. 2020. Privacy-Utility Tradeoffs in Routing Cryptocurrency over Payment Channel Networks. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 2, Article 29 (June 2020), 40 pages. https://doi.org/10.1145/3392147

1 INTRODUCTION

As the adoption of cryptocurrencies grows to unprecedented levels [4], the scalability limitations of these technologies have become apparent. For example, Bitcoin today can process up to seven transactions per second with a confirmation latency of hours [13, 39]. For comparison, the Visa network can process tens of thousands of transactions per second with a confirmation latency of

Authors' addresses: Weizhao Tang, Carnegie Mellon University, USA, wtang2@andrew.cmu.edu; Weina Wang, Carnegie Mellon University, USA, weinaw@cs.cmu.edu; Giulia Fanti, Carnegie Mellon University, USA, gfanti@andrew.cmu.edu; Sewoong Oh, University of Washington, USA, sewoong@cs.washington.edu.

2476-1249/2020/6-ART29 $15.00
https://doi.org/10.1145/3392147

seconds [35]. This gap raises questions about whether cryptocurrencies are fundamentally able to support as much traffic as traditional, centralized solutions.

In response to these challenges, several cryptocurrencies have turned to a class of scalability solutions called *payment channel networks* (PCNs) [28]. The core idea is that instead of committing every transaction to the blockchain, a separate overlay network (the PCN) is maintained, in which each node represents a user, and each edge represents pre-allocated funds that can be efficiently and quickly transacted between the two endpoints of the edge under a mutual agreement. Critically, those transactions on the PCN are committed to the blockchain only in periodic batches, which reduces the frequency with which users must call the (slow, inefficient) blockchain consensus mechanism. Users can send money to non-adjacent nodes on the PCN by relaying money through intermediate nodes on the graph. PCNs are viewed in the blockchain community as one of the most promising scalability solutions today [16, 34], and several major cryptocurrencies are staking their long-term scaling plans on PCNs. Prominent examples include Bitcoin's Lightning network [28] and Ethereum's Raiden network [2].

Despite this excitement, there remain several technical challenges. Principal among them is a privacy-preserving routing problem: each time users wish to route a transaction, they must find a path through the PCN with enough pre-allocated funds to route the transaction. However, in today's PCNs, edge balances are not publicly revealed for privacy reasons, making it difficult to find such a route reliably. The goal of this paper is to study privacy-utility tradeoffs that arise in such a PCN transaction routing. Before defining the problem more precisely, we begin with a brief primer.

1.1 Payment Channel Networks (PCNs)

A *payment channel* is a transaction between two parties that escrows currency for use only between those two parties for some amount of time. For example, Alice and Bob could escrow 4 and 2 tokens, respectively, for the next week. The escrow transaction is committed to the blockchain to finalize it. Once the channel is finalized, Alice and Bob can send escrowed funds back and forth by digitally signing the previous state of the channel and the new updated transaction. For example, Alice can send 3 of her 4 tokens to Bob, so that the new channel state is (Alice=1, Bob=5). Once the parties decide to close the channel, they can commit its final state through another blockchain transaction. Cryptographic and incentive-based protections prevent users from stealing funds in a channel, e.g., by committing an outdated state. Maintaining a payment channel has an opportunity cost because users must lock up their funds while the channel is active, and they are not actually paid until the channel is closed. Hence, it is not practical to expect users to maintain a channel with every individual with whom they may ever need to transact.

A *payment channel network* (PCN) gets around this problem by setting up a graph of bidirectional payment channels. The key idea is that if Alice wants to transact with Charlie, but is only connected to him via Bob, then Bob can act as a relay for Alice's money, passing it along to Charlie. Again, cryptographic protections are used to ensure that Bob does not steal Alice's relayed money, but he does receive a small fee as payment for his cooperation. Notice that if Alice wants to send r tokens to any node in the network, she must first find a directed path to that node with at least r tokens on every (directed) edge. For example, in Figure 1, suppose Alice wants to send $r = 3$ tokens to Charlie. She has two paths available: $A \to B \to C$ and $A \to D \to E \to C$. However, the edge $E \to C$ has only 2 tokens to send in that direction, so it cannot support Alice's transaction. Hence, this transaction fails. Suppose Alice instead sends her transaction over the first path, $A \to B \to C$. After her transaction is processed, each of the channels moves $r = 3$ tokens (plus a small processing fee) from one side of the channel to the other. Note that one could also packet-switch transactions, so Alice sends part of her transaction over multiple paths. In today's PCNs, this functionality is not yet

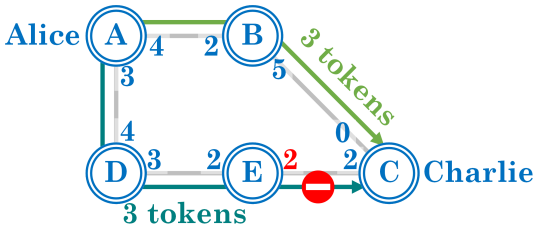

Fig. 1. Payment channel network. Alice wants to send 3 tokens to Charlie. Transaction fees are omitted for simplicity.

implemented [1]; however, such routing schemes can give significant performance enhancements [10, 18, 27, 32]. Packet-switched routing is beyond our scope.

Our work studies the act of finding a path from source to destination with sufficient balance to support a transaction. If the full graph and edge weights were known, finding such a path would be straightforward (albeit computationally-intensive). However, if users were given access to instantaneous channel balances of the whole network, any passive observer could trivially see that, for instance, r tokens flowed along a path from Alice to Charlie. This is a serious breach of privacy.

Today's PCNs do not reveal instantaneous balance information in an attempt to prevent observers from inferring other users' transaction patterns. Instead, users are given access only to the graph topology, and the *sum* of balances in either direction on each channel, which remains constant for the duration of the channel. Because of this design decision, PCN users are forced to guess if a given path has enough balance to support a particular transaction by attempting to send their transaction over that path. If it fails, the user tries another path until the transaction either completes or times out; upon a timeout, the user can instead process her transaction on the blockchain, which is comparatively slow and expensive. This guess-and-check routing approach uses unnecessary resources and severely limits the success rates of today's PCNs [37]. Every time a user tries to route over a path that will ultimately fail, it temporarily ties up funds along that path, which cannot be used by other transactions. Moreover, since users are forced to guess channel balances, they are more likely to not find any valid path prior to transaction timeout. Our goal in this work is to understand whether privacy must always come at such a high cost. In particular, we consider whether a system could reveal *noisy* channel balances in an effort to gracefully trade off privacy for utility.

1.2 Contributions

We make four main contributions:

- We theoretically model the routing problem in PCNs and define distribution-free metrics for privacy and utility. In particular, we relate the *success rate* of a scheme, or the fraction of successfully routed transactions, to a simplified but analytically tractable quantity we call *utility*.
- We show a restrictive, so-called *diagonal* upper bound on the privacy-utility tradeoff for these metrics over general graphs and a significant class of shortest-path transaction routing strategies. We show that the diagonal bound is tight by designing noise mechanisms that achieve it.
- The diagonal bound is a somewhat negative result, suggesting that a good tradeoff is not possible. However, we show that by relaxing certain assumptions (e.g., the shortest-path routing assumption), we can break the diagonal barrier. Indeed, one can design noise mechanisms that asymptotically achieve a perfect privacy-utility tradeoff. However, this comes at the cost of increasingly long paths, i.e., increasingly expensive routing fees.

- We demonstrate through simulation that even if one were to consider an average-case utility metric (fraction of successful transactions, or success rate) rather than a worst-case one, the privacy-success rate tradeoff is still not favorable for shortest-path routing.
  Overall, our simulations suggest that trading off privacy for utility does not give significant gains unless the system operates either in a low-privacy regime or low-utility regime; today's PCNs operate in the low-utility regime.

In sum, our results suggest that PCNs may not be able to provide utility and privacy simultaneously. Moreover, our theoretical analysis is conducted under an adversarial model that (a) is passive, and (b) does not exploit temporal correlations in transaction patterns. Hence, actual privacy threats are likely even more dire than our results indicate. PCN system designers may therefore need to make an explicit choice regarding whether the value of PCNs comes mainly from their potential for improving performance or privacy, and choose an operating point accordingly.

2 MODEL

We model the PCN as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the participating nodes with $n = |\mathcal{V}|$, and $\mathcal{E}$ the set of edges, or payment channels, in the PCN. Each edge $(u, v) \in \mathcal{E}$ is associated with two weights, b_{uv} and b_{vu}, which denote the balances from u to v and from v to u, respectively. The *capacity* of the channel, denoted as $C_{uv} = b_{uv} + b_{vu}$, is assumed to be a constant; this models a setting where the channel remains open for the entire duration of the experiment. Recall that our goal is to release noisy channel balances, so the true balances may not be equal to the publicly-released channel balances, which we denote by $\tilde{b}_{uv}$ and $\tilde{b}_{vu}$, respectively.

We assume an arbitrary sequence of transactions $x_1, x_2, \ldots$ enters the system sequentially. Each transaction x_i has an associated source $s(x_i)$, destination $d(x_i)$, amount $r(x_i)$, and timestamp $t(x_i)$, where $t(x_i) < t(x_{i+1})$ for all i. Each transaction is processed instantaneously at its time-of-arrival timestamp. That is, we do not account for concurrent transactions. For a path P on the graph, we use $s(P)$ and $d(P)$ to denote its source and destination, respectively. At transaction x's time of arrival, $t(x)$, the source $s(x)$ chooses a path P from the network with source $s(P) = s(x)$ and destination $d(P) = d(x)$, such that P appears to have enough balance according to the publicly visible balances. That is, $\forall (u, v) \in P$, $\tilde{b}_{uv} \geq r(x)$. Recall that channel balances are noisy, so a path that appears to have enough balance may not in reality.

If the path chosen for transaction x does not have enough balance (i.e., $\exists (u, v) \in P : b_{uv} < r(x)$), the transaction fails. In this case, both the visible channel balances $\tilde{b}_{uv}$ and the true channel balances b_{uv} remain unchanged. If the path chosen for transaction x has enough balance (i.e., $\forall (u, v) \in P, b_{uv} \geq r(x)$), the transaction succeeds, and $\forall (u, v) \in P$, the true channel balance is updated as $b_{uv} = b_{uv} - r(x)$ and $b_{vu} = b_{vu} + r(x)$. The visible channel balances, on the other hand, are updated according to a *noise mechanism* (or *mechanism*). The noise mechanism is probabilistic and chosen by the system designer; given an input path P, it outputs a random set of edges $Q \subseteq P$ such that $\forall (u, v) \in Q$, the public balance $\tilde{b}_{uv}$ is updated to the true new balance, b_{uv}. We denote this conditional probability distribution by $\mathbb{D}[Q|P]$.

Notice two things about this model: first, we never update public balances on edges that were not involved in a transaction, i.e., edges that are not elements of path P. This modeling choice was made in part for analytical tractability and in part for practicality; since nodes anyway must communicate with all relays in P, it is easy to include an instruction for the noisy balance update. Reaching other nodes would require additional communication that may not be practical. Second, note that if an edge is updated, it is only updated to its true balance—never a noisy version of its balance. One could additionally impose the condition that after choosing a subset Q, the noise mechanism updates the balances in Q by different amounts (not necessarily equal to $r(x)$); however,

as we will discuss in Section 2.1, our privacy metric only considers adversaries who aim to identify the source or destination of a transaction—not the amount. If an adversary observes *any* update on an edge, it knows that edge was involved in the transaction; knowing the exact amount does not give any more information about the source or destination. Hence, to maximize utility, we might as well reveal the full balance update.

2.1 Privacy Metric

Our adversary is an honest-but-curious user that passively observes the network and tries to infer the source and destination of the first transaction x to pass through the system; the adversary does *not* try to guess the transaction amount $r(x)$. We make this modeling choice for simplicity and because transaction patterns often matter more than transaction amounts; for instance, the fact that one sends money to an abortion clinic or to a political organization can be more telling than the actual amount transferred. We assume that the initial public balances are all identical to the real balances, i.e., $\forall (u,v) \in \mathcal{E}$, $b_{uv} = \tilde{b}_{uv}$ at time 0, and all the balances in the PCN are sufficient to support x; hence, this metric is equivalent to considering an arbitrary transaction x but giving the adversary knowledge of the true balances shortly before $t(x)$. More precisely, once x has been processed, the adversary guesses one node $v \in \mathcal{V}$ with probability $\mathbb{A}[v|Q]$, where $\mathbb{A}$ is a randomized adversarial strategy. If $v \in \partial P$, where $\partial P \triangleq \{s(P), d(P)\}$, i.e., if the adversary guesses the source *or* destination correctly, it wins. To avoid making assumptions about the transaction or PCN distribution, we consider a worst-case metric over both. As shown in (2.1), privacy is defined as the minimax probability that the adversary makes a wrong estimate:

$$\begin{aligned}\Pi(\mathbb{D}) &= 1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \sum_{v \in \partial P, Q \subseteq P} \mathbb{D}\left[Q|P\right] \mathbb{A}\left[v|Q\right] \\ &= 1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \sum_{Q \subseteq P} \mathbb{D}\left[Q|P\right] \mathbb{A}\left[\partial P|Q\right], \end{aligned} \tag{2.1}$$

where $\mathcal{P}$ denotes the set of paths in $\mathcal{G}$. For convenience, we let $\mathbb{A}[\partial P|Q]$ denote $\sum_{v \in \partial P} \mathbb{A}[v|Q]$.

Notice that $\mathbb{A}[\cdot|Q]$ is a probability distribution with event space $\mathcal{V}$, i.e. the supremum over $\mathbb{A}$ is taken in the space

$$\mathcal{A} = \left\{ \mathbb{A} \middle| \forall u \in \mathcal{V}, P \in \mathcal{P}, Q \subseteq P : \mathbb{A}[u|Q] \geq 0, \sum_{v \in \mathcal{V}} \mathbb{A}[v|Q] = 1 \right\}.$$

So constraint $\mathbb{A} \in \mathcal{A}$ is an intersection of linear constraints. In addition, the objective of (2.1) is the minimum of finite number of affine functions of $\mathbb{A}$. Hence, the optimization problem in (2.1) is equivalent to an LP. Because $\mathcal{A}$ is compact, the supremum of the optimization problem is attained. However, solving this optimization is intractable in general, due to the optimization over all the paths in the PCN. We will show that under certain restrictions, a solution can be computed efficiently.

2.2 Utility Metric

The performance of a PCN is commonly measured by its *success rate*, or the fraction of transactions it successfully processes [24, 30, 32]. However, expected success rate is a complicated function that depends on transaction workloads, graph topology, and initial balances. Since we wish to avoid making assumptions on these characteristics, we instead consider a simpler notion of utility: how representative are the observed balances of the true underlying balances? Specifically, given the true balance of a link, we consider the probability that the observed balance is equal to the true balance, which we will refer to as the *truthful probability*. In general, this probability for a given

PCN still depends on the transaction workload, as well as parameters like the path length. Hence, we consider a *worst-case* transaction workload that minimizes the truthful probability.

To this end, we define the quantity in (2.2) below as the utility of mechanism $\mathbb{D}$. It equals the minimum probability, over all paths and edges in paths, of a public edge balance equaling the true edge balance after a transaction passes through it:

$$U(\mathbb{D}) = \min_{P \in \mathcal{P}} \min_{\varepsilon \in P} \sum_{Q: Q \ni \varepsilon, Q \subseteq P} \mathbb{D}\left[Q|P\right] \tag{2.2}$$

In the proposition below, we show how this notion of utility is related to the truthful probability.

Proposition 2.1. *The truthful probability of any edge given any value of the true balance is lower bounded by the utility $U(\mathbb{D})$.*

Proof. We focus on a particular edge $\varepsilon \in \mathcal{E}$. Assume *at the current state*, the true and public balances of ε are B_ε, $\tilde{B}_\varepsilon$, respectively. Define event

$$E_\varepsilon \triangleq \text{there exists a previous transaction going through } \varepsilon.$$

If $\neg E_\varepsilon$ happens, then $\mathbb{P}[B_\varepsilon = \tilde{B}_\varepsilon | \neg E_\varepsilon] = 1$, since both balances are identical to their initial states, at which time all public balances are truthful.

If E_ε happens, we focus on the state when the last transaction T_{-1} on path P' involving ε was about to be launched. At this state, we assume the true and public balances of ε are b_ε and $\tilde{b}_\varepsilon$, while they have been B_ε and $\tilde{B}_\varepsilon$ since T_{-1} was launched, respectively. By definition of utility,

$$U(\mathbb{D}) \leq \sum_{Q: Q \ni \varepsilon, Q \subseteq P'} \mathbb{D}\left[Q|P'\right] = \mathbb{P}[\tilde{B}_\varepsilon = B_\varepsilon | E_\varepsilon, T_{-1}].$$

Since T_{-1} could be arbitrary, $\mathbb{P}[\tilde{B}_\varepsilon = B_\varepsilon | E_\varepsilon] \geq U(\mathbb{D})$.

To summarize, $\mathbb{P}[B_\varepsilon = \tilde{B}_\varepsilon] \geq U(\mathbb{D})$ holds for all $\varepsilon \in \mathcal{E}$ regardless of previous workload on the network. □

From the proof of Proposition 2.1, we can also see that $U(\mathbb{D})$ is *equal* to the truthful probability of an edge ε in the following scenario: a transaction happens along a path P' that contains ε, and ε and P' are the minimizers in the definition of $U(\mathbb{D})$. Therefore, $U(\mathbb{D})$ characterizes the truthful probability in a worst-case sense. For this reason, we will focus on $U(\mathbb{D})$ as our utility metric.

A natural question is how this utility metric relates to the success rate of a PCN. The answer is complicated, depending on a number of factors. However, we show in Section 5 that under certain workloads and network conditions, our utility metric and the success rate of a PCN appear to be monotonically related. Hence, maximizing utility is as good as maximizing success rate.

3 FUNDAMENTAL LIMITS

As system designers, we want the the privacy and utility metrics defined in (2.1) and (2.2) to be high, i.e., close to 1. In this section, we show that for an important class of routing algorithms, this is not possible; additionally, we provide a tight upper bound on the tradeoff between these quantities for general graph topologies.

Let us begin by considering two extreme points. First, consider a setting with perfect utility, $U(\mathbb{D}) = 1$. To achieve perfect utility, on every path, the balance of every edge must be truthfully updated. Hence, if the adversary simply guesses the source of the observed updated path, it always wins, so we have no privacy: $\Pi(\mathbb{D}) = 0$.

Next, consider a case with no utility, $U(\mathbb{D}) = 0$. This implies that for every path, noise mechanism $\mathbb{D}$ always hides the balance update on at least one edge from the true path. Regardless of the noise mechanism, the adversary can always pick a node uniformly at random. Since there are n nodes, it

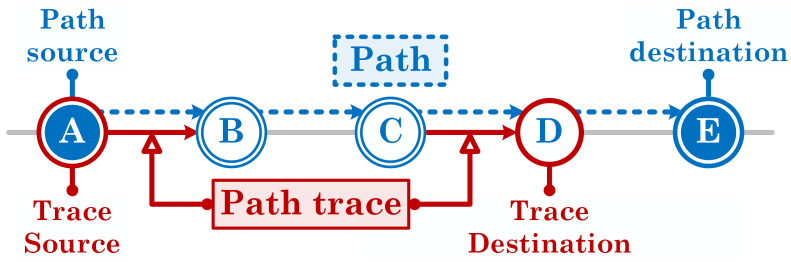

Fig. 2. Example of a path from A to E and a corresponding path trace, the set of edges $A \longrightarrow B$ and $C \longrightarrow D$.

guesses the source or destination with probability $2/n$, so $\Pi(\mathbb{D}) \leq 1 - \frac{2}{n}$, $\forall \mathbb{D}$. Hence, we have an upper bound on the privacy metric at the two extremes of the utility spectrum.

The more challenging and interesting case arises when $0 < U(\mathbb{D}) < 1$. We first define a *path trace*, which intuitively describes a (possibly disjoint) set of edges that are elements of a path between two endpoints. This set of edges must include the first and last hop of the path, adjacent to the two endpoints (Figure 2).

Definition 3.1. Given two nodes x and y on an arbitrary undirected graph $(\mathcal{V}, \mathcal{E})$ with set of available paths $\mathcal{P}$, a set of oriented edges Q is a **path trace** from x to y, if and only if

(1) x has an incident edge in Q with x being the source;
(2) y has an incident edge in Q with y being the destination;
(3) there exists a path $P \in \mathcal{P}, P \supseteq Q$ from x to y.

We say x is a **source** of Q, and y is a **destination** of Q.

In other words, a path trace is what the adversary sees after the noise mechanism updates a subset of edges in a given path. A path trace is always a subset of an available path in $\mathcal{P}$. Our first main result, Theorem 3.2, states that if $\mathcal{P}$ includes only shortest paths between nodes, then the privacy metric is upper bounded by a linear relation that we call the *diagonal bound*. Notice that shortest-path routing is extremely common. First, today's PCNs like the Lightning network implement shortest-path routing by default [1]. Second, since intermediate relays in PCNs extract transaction fees, shortest-path routing can be thought of as a proxy for cheapest routing, which users are incentivized to use. Hence this result applies to the vast majority of routes used in practice.

Theorem 3.2 (The Diagonal Bound). *On a network $(\mathcal{V}, \mathcal{E}, \mathcal{P})$ with $n = |\mathcal{V}| \geq 2$, if $\mathcal{P}$ includes only shortest paths, then for any noise mechanism $\mathbb{D}$, its privacy $\Pi(\mathbb{D})$ and utility $U(\mathbb{D})$ satisfy*

$$\Pi(\mathbb{D}) \leq \left(1 - \frac{2}{n}\right)[1 - U(\mathbb{D})]. \tag{3.1}$$

Notice that this bound, plotted in Figure 3, does not make assumptions about the structure of the underlying graph. Intuitively, the result holds because for any observed path trace taken from a shortest path between two nodes on an arbitrary graph, the adversary can uniquely identify its source and destination. For example, consider Figure 4 (left). On a tree, even if the noise mechanism reveals only a subset of edges in a path, the adversary can trivially reconstruct the interior path exactly. From this reconstruction, the endpoints are clear. Notice that in a path trace, we know the orientation of each edge because a truthfully-updated channel will always reveal the direction of money flow. In Figure 4 (right), we see that on a grid graph, the adversary cannot always reconstruct the exact path, as there may be multiple shortest paths consistent with the path trace.

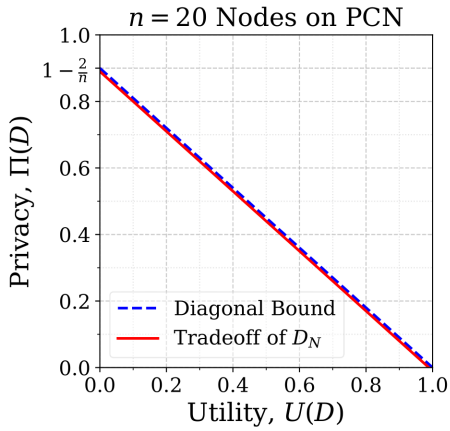

Fig. 3. The diagonal upper bound of Theorem 3.2, and the achievable tradeoff for the all-or-nothing scheme. The all-or-nothing curve is slightly displaced from the upper bound for greater visibility.

For instance, the adversary cannot tell if $B \to C \to E$ or $B \to D \to E$ was the true path. However, the adversary *can* uniquely determine the set of endpoints of the trace: A and E. Any other choice (e.g, A and D) that passes through the full set of path trace edges has a strictly longer path length. This observation is true for general graphs, allowing the adversary to use the source or destination of the path trace as its estimate of the source or destination of the original path. This in turn causes the privacy metric for a noise mechanism to be governed primarily by its behavior near the source and destination, giving rise to the diagonal bound.

We next prove that the bound in Theorem 3.2 is tight. The proof of the upper bound implies that hiding edges in the interior of a path does not improve privacy if one or both of the endpoints of the path are revealed. In other words, an adversary cannot always determine the sequence of edges between the endpoints of a path trace, but it can identify the *endpoints* of the trace. Because there is no privacy benefit to hiding interior edges of a path, we should maximize utility (for a

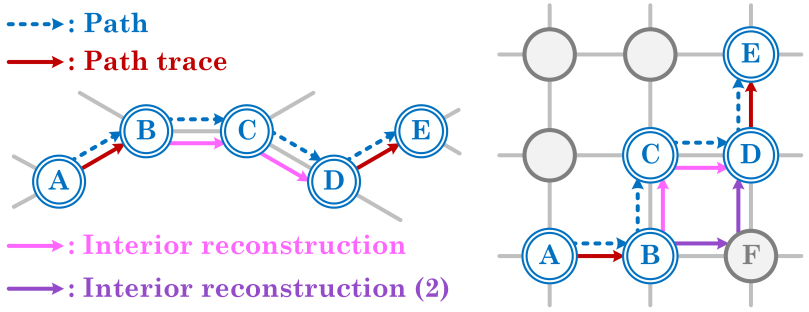

Fig. 4. Examples of a path and a path trace on a 3-regular tree (left) and a grid graph (right), assuming shortest-path routing. For a tree, given any path trace, we can always uniquely determine the path that generated the path trace (up to the endpoints of the path trace). On a grid graph (indeed, on general graphs), we can uniquely determine the endpoints of the path trace, but not necessarily the full path.

given privacy level) by revealing all interior edges. This motivates the so-called *all-or-nothing* noise mechanism.

Definition 3.3. For a transaction routed over path P, the **all-or-nothing noise mechanism** $\mathbb{D}_{\mathbf{N}}$ either truthfully updates balance on every edge of P with probability $U(\mathbb{D}_{\mathbf{N}})$, or updates nothing with probability $1 - U(\mathbb{D}_{\mathbf{N}})$.

We define a *reachable* network as follows:

Definition 3.4. A network $(\mathcal{V}, \mathcal{E}, \mathcal{P})$ is **reachable**, if for any pair of nodes $(u, v) \in \mathcal{V}^2$ with $u \neq v$, there exists a path $P \in \mathcal{P}$ that goes from u to v, or from v to u.

Note that reachable networks are more restrictive than the notion from the literature of *strongly connected networks*, because we limit the set of usable paths to $\mathcal{P}$.

Lemma 3.5. *On a reachable network,*

$$\sup_{\mathbb{A}[\cdot|\varnothing]} \min_{P \in \mathcal{P}} \mathbb{A}[\partial P|\varnothing] = \frac{2}{n}. \tag{3.2}$$

Proof. Because $\mathbb{A}[\cdot|\varnothing]$ is a probability distribution, for any strategy $\mathbb{A}$, there exist 2 nodes u_1, u_2 such that

$$\mathbb{A}[u_1|\varnothing] + \mathbb{A}[u_2|\varnothing] \leq \frac{2}{n}.$$

By assumption that the network is reachable, there exists a path P_u connecting u_1, u_2. Hence,

$$\min_{P \in \mathcal{P}} \mathbb{A}[\partial P|\varnothing] \leq \mathbb{A}[\partial P_u|\varnothing] \leq \frac{2}{n}, \ \forall \mathbb{A}.$$

In the meantime, the adversary who guesses each node uniformly could make the inequality above tight, which proves (3.2). □

On a reachable network, the below theorem characterizes the privacy-utility tradeoff for the all-or-nothing noise mechanism.

Theorem 3.6. *On a reachable network, the privacy-utility tradeoff of the all-or-nothing noise mechanism is characterized by*

$$\Pi(\mathbb{D}_{\mathbf{N}}) = \left(1 - \frac{2}{n}\right)[1 - U(\mathbb{D}_{\mathbf{N}})]. \tag{3.3}$$

Although this result holds for all reachable networks, regardless of routing policy, it also implies that the upper bound in Theorem 3.2 is tight for shortest-path routing over reachable networks.

Proof. When the entire path is updated, the adversary $\mathbb{A}_0$ may directly pick an endpoint of the observed path as its estimate. Hence,

$$\sup_{\mathbb{A}} \min_{P' \in \mathcal{P}} \mathbb{A}[\partial P'|P'] = \mathbb{A}_0[\partial P|P] = 1, \quad \forall P \in \mathcal{P}.$$

By our privacy definition,

$$\begin{aligned}
\Pi(\mathbb{D}_{\mathbf{N}}) &= 1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \sum_{Q \subseteq P} \mathbb{D}_{\mathbf{N}}[Q|P]\mathbb{A}[\partial P|Q] \\
&= 1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \{[1 - U(\mathbb{D})]\, \mathbb{A}[\partial P|\varnothing] + U(\mathbb{D})\mathbb{A}[\partial P|P]\} \\
&= [1 - U(\mathbb{D})] \left(1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \mathbb{A}[\partial P|\varnothing]\right)
\end{aligned}$$

$$= [1 - U(\mathbb{D})] \left(1 - \frac{2}{n}\right). \quad (*)$$

Obtained by Lemma 3.5, (*) justifies the original claim. □

The results of this section are not encouraging; they imply that a perfect privacy-utility tradeoff is impossible as long as nodes use shortest-path routing. Since there is a monetary cost associated with taking longer paths, most users are likely to default to shortest-path routing. Nonetheless, some users do care about privacy and may be willing to pay for it [3, 5]. The next section explores various techniques for breaking the diagonal bound, and the costs associated with doing so.

4 BREAKING THE DIAGONAL BARRIER

The goal of this section is to break the diagonal barrier imposed by Theorem 3.2 by relaxing two key assumptions: one related to adding uncertainty to the candidate endpoint set, and the other related to adding uncertainty to the set of candidate paths. More precisely, Theorem 3.2 makes two main assumptions: (i) utility $U(\mathbb{D})$ is computed as a minimum over all network edges, as in the definition (2.2), and (ii) all transactions are routed using a shortest path between the source and destination. We will relax the first assumption to introduce endpoint uncertainty, and the second to introduce path uncertainty.

Assumption (i): Definition (2.2) of utility may not be appropriate when the PCN includes endpoint nodes with degree one, or clients that connect to "gateway" routers. This could happen, for example, if PCNs evolve so that merchants run the majority of router nodes, which maintain several well-funded channels, and end users connect to merchants only for their own transactions, i.e., without participating in transaction relaying. As an extreme example, consider a star network, which is a tree of depth one. Any routing of a transaction requires at most two edges, whose balances are already known to the nodes directly involved in the transaction, i.e., the sender and the receiver. Because the sender and receiver already know the balances on their adjacent edges, they could transmit this information out of band, so hiding the balance of any edge does not decrease success rate, but provides perfect privacy against an external observer. In Section 4.1, we show how to improve the privacy-utility tradeoff by allowing the source and destination to exchange their adjacent true balances prior to transacting.

Assumption (ii): The second assumption might not hold if users are willing to route transactions over paths longer than the shortest path(s). At the cost of incurring more fees, such longer routing can intuitively achieve better privacy-utility tradeoff. Although the gains are difficult to quantify in general, we precisely quantify them for a special network topology in Section 4.2.

4.1 Endpoint Uncertainty

In real-world PCNs, users know the true balances (or weights) on their directly adjacent channels (or edges). For example, when a user Alice tries to transact with neighboring user Bob, Alice and Bob already know each other's true balances. Therefore, for any direct transaction among neighbors, success rate is not sacrificed even if the balances are kept private. In an extension of this idea, consider a PCN model with *users* and *servers*; users make transactions among themselves, whereas servers relay them. Servers are connected in a network structure $(\mathcal{V}_S, \mathcal{E}_S)$, whereas each user node is connected to one server node. Let $\mathcal{V}_U, \mathcal{E}_U$ denote the set of user nodes and user-server channels, respectively. Then there exist functions $J : \mathcal{V}_U \to \mathcal{V}_S$ and $K : \mathcal{V}_U \to \mathcal{E}_U$, where for a user node $v \in \mathcal{V}_U$, $J(v)$ is the server to which v is connected, and $K(v)$ is its unique incident channel, which connects v and $J(v)$. Furthermore, let $(n_S, n_U) = (|\mathcal{V}_S|, |\mathcal{V}_U|)$. The public balances on the user-server channels are never updated, while those on the inter-server channels are updated with a privatizing noise mechanism of our choice.

Under the following assumption that the set $\mathcal{P}$ of available paths are not too restricted, we show that just privatizing the inter-server channels is sufficient to achieve improved privacy-utility tradeoff.

Assumption 1. *For any oriented path $P \in \mathcal{P}$ from x to y,*

- *if $x \notin \mathcal{V}_S$, then $P - \{(x, J(x))\} \in \mathcal{P}$;*
- *if $x \in \mathcal{V}_S$, then $\forall v : J(v) = x$, we have $\{(v, x)\} \cup P \in \mathcal{P}$);*
- *if $y \notin \mathcal{V}_S$, then $P - \{(J(y), y)\} \in \mathcal{P}$;*
- *if $y \in \mathcal{V}_S$, then $\forall v : J(v) = y$, we have $\{(y, v)\} \cup P \in \mathcal{P}$).*

Assumption 1 states that if an oriented path from server x' to server y' is observed, the adversary cannot obtain more information from $\mathcal{P}$ about the source or destination. For example, he only learns that the source can be server x' itself or one of its incident users.

Proposition 4.1. *Suppose we have a user-server network $(\mathcal{V}_S \cup \mathcal{V}_U, \mathcal{E}_S \cup \mathcal{E}_U)$, and the set $\mathcal{P}$ of available paths satisfy Assumption 1, the server sub-network $(\mathcal{V}_S, \mathcal{E}_S)$ is reachable under $\mathcal{P}$, and all the transactions are routed via the shortest paths. When the channels on the server network $(\mathcal{V}_S, \mathcal{E}_S)$ have balances governed by the all-or-nothing noise mechanism $\mathbb{D}_{\mathbf{N}}$ with utility $U(\mathbb{D}_{\mathbf{N}}) = \alpha$, and the channels on the user network $(\mathcal{V}_U, \mathcal{E}_U)$ are completely hidden, the privacy metric $\Pi(\mathbb{D}_{\mathbf{N}})$ is*

$$\Pi(\mathbb{D}_{\mathbf{N}}) = \left(1 - \frac{2}{n_S + n_U}\right)(1 - \alpha) + \frac{\mu}{\mu + 1}\alpha \,, \tag{4.1}$$

where μ is a lower bound on the number of users attached to any server node.

Compared with all-or-nothing noise on a full PCN (Theorem 3.6), the privacy level of the user-server model is higher by $\mu U(\mathbb{D})/(1 + \mu)$. Figure 7, left panel, plots the gains achievable from Theorem 4.1 under all-or-nothing noise. As the number of users per server increases, we get improved tradeoffs, eventually achieving maximum privacy with no utility loss in the limit as $\mu, n_S, n_U \to \infty$. Intuitively, the adversary cannot distinguish between sources or destinations connected to the first and last server nodes. Since the adjacent user nodes as well as the last server node can all be the real destination of money flow, the adversary's best strategy is to guess uniformly at random. Hence the gain in privacy comes from the increased uncertainty at the end nodes, which does not sacrifice the success rate of routing transactions. This observation can be generalized to each user having multiple connections to servers, as long as the users do not relay transactions. Moreover, preliminary evidence suggests that user-server models may naturally emerge in practice; thus far, we have seen the emergence of routing hubs on the Lightning network, complemented by many users with few channels (Figure 17).

Proposition 4.1 holds when each user is allowed to be connected to only one server. In this case, the size of server sub-network is limited, since $n_S \leq n/(\mu + 1)$. We can easily extend this result to the case where each user node is allowed to share channels with multiple server nodes, but users do not relay transactions. For example, in Figure 5, the light-green user is allowed to maintain two channels with two different servers. In this case, the network structure will be much more compact, and its privacy is only slightly worse than (4.1), as seen in the following:

Proposition 4.2. *Assume the assumptions of Proposition 4.1 except that each user may connect to multiple servers (but not relay transactions). The privacy metric $\Pi(\mathbb{D}_{\mathbf{N}})$ of the all-or-nothing noise $\mathbb{D}_{\mathbf{N}}$ satisfies*

$$\Pi(\mathbb{D}_{\mathbf{N}}) \geq \left(1 - \frac{2}{n_S + n_U}\right)(1 - \alpha) + \frac{\mu}{\mu + 2}\alpha \,, \tag{4.2}$$

where μ is a lower bound on the number of user nodes attached to any server node.

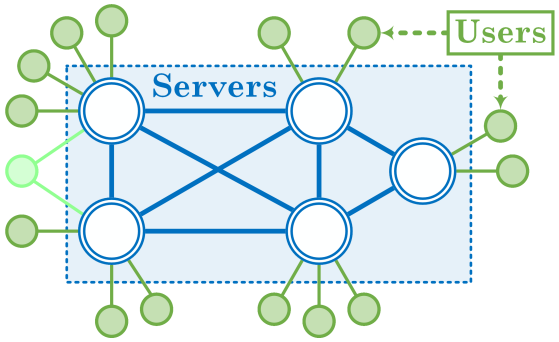

Fig. 5. The user-server model consists of server nodes that route transactions, and user nodes who make transactions, but do not relay other nodes' transactions.

This suggests that in practice, user endpoints can improve their privacy by not updating their balances without losing any utility, as long as they have an out-of-band channel of communication with the recipient of a transaction. This may not always be feasible, e.g., in cases where the sender or recipient wishes to remain anonymous.

4.2 Path Uncertainty

The previous subsection showed how a special class of networks can be exploited to increase user privacy by introducing uncertainty regarding the endpoints of a path under all-or-nothing noise. In this section, we explore techniques for obfuscating the transaction path itself. When users are allowed to use longer paths, the noise mechanism can hide edges so that path traces do not reveal the source/destination pair. This is not true under the shortest path routing as illustrated in Section 3. For the rest of this section, we consider the special case of a *complete graph*. Although complete graphs are not realistic for deployed PCNs, they provide a method for analyzing the privacy definition LP in (2.1), as well as inspiration for noise mechanisms that can break the diagonal barrier in Sec. 3.

Definition 4.3. Given the edges on a directed path P, let the *odd* edges include the 1st, 3rd, 5th, ... edges and the *even* group include the 2nd, 4th, 6th, ... edges. The **alternating noise mechanism** $\mathbb{D}_{\mathbf{T}}$ is defined as follows:

- At utility $\alpha \in [0, 0.5]$, $\mathbb{D}_{\mathbf{T}}$ either updates the balances of the odd edges with probability α, updates balances of the even edges with same probability, or updates nothing with probability $1 - 2\alpha$.
- At utility $\alpha \in (0.5, 1]$, $\mathbb{D}_{\mathbf{T}}$ either updates the balances of the odd edges with probability $1 - \alpha$, updates balances of the even edges with same probability, or updates nothing with probability $2\alpha - 1$.

To preserve symmetry, $\mathcal{P}$ is assumed to be the set of all simple paths of fixed length L. Note that it may be computationally infeasible to find a simple path of given length that satisfies a balance constraint in practice. Intuitively, we expect alternating noise to have good privacy; even when the adversary knows the path length L, each revealed edge could be the first or last edge, and the graph topology gives no way to differentiate between edges. For example, in Figure 6, the adversary is able to discern that the even edges were revealed since $L = 5$ and only two edges are revealed. Hence it must guess the source/destination uniformly from the nodes that are not part of the path trace. If the odd edges had been revealed, then because the graph is complete, any ordering of the edges would have been feasible. Hence, the adversary would need to guess one of the endpoints of

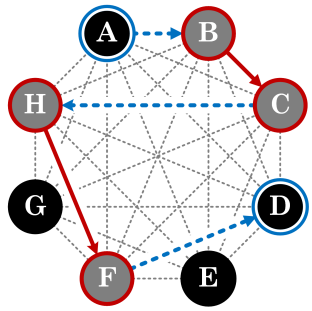

Fig. 6. A path and path trace for the alternating noise mechanism; in this case, even edges were revealed.

the revealed edges uniformly at random. Combined, these cases give asymptotically perfect privacy as $L, n \to \infty$. The next theorem characterizes this tradeoff precisely.

Theorem 4.4. *Suppose $L \geq 2$ and the alternating scheme $\mathbb{D}_{\mathbf{T}}$ has utility $\alpha = U(\mathbb{D}_{\mathbf{T}})$ and privacy $\Pi(\mathbb{D}_{\mathbf{T}})$. If $2 \mid L$, i.e. L is even, its privacy-utility tradeoff on a complete graph is characterized by*

$$\Pi(\mathbb{D}_{\mathbf{T}}) = \begin{cases} 1 - \frac{2}{n} - \left(\frac{2}{\min\{L, n-L\}} - \frac{4}{n}\right)\alpha, & 0 \leq \alpha \leq \frac{1}{2}; \\ \left(2 - \frac{2}{\min\{L, n-L\}}\right)(1-\alpha), & \frac{1}{2} < \alpha \leq 1. \end{cases} \tag{4.3}$$

If $2 \nmid L$, i.e. L is odd, its privacy-utility tradeoff is characterized by

$$\Pi(\mathbb{D}_{\mathbf{T}}) = \begin{cases} 1 - \frac{2}{n} - \left(\frac{2}{L+1} + \frac{2}{n-L+1} - \frac{4}{n}\right)\alpha, & 0 \leq \alpha \leq \frac{1}{2}; \\ \left(2 - \frac{2}{L+1} - \frac{2}{n-L+1}\right)(1-\alpha), & \frac{1}{2} < \alpha \leq 1. \end{cases} \tag{4.4}$$

This tradeoff is plotted in Figure 7 (right) for two path lengths, $L = 9$ and $L = 60$. Notice that the difference between these curves is small, even though the difference in path lengths is almost an order of magnitude. Hence, using longer paths appears to have diminishing returns for alternating noise.

Another natural mechanism which requires less coordination among the nodes in a path is the *i.i.d. noise mechanism*.

Definition 4.5. The **i.i.d. noise mechanism** $\mathbb{D}_{\mathbf{I}}$ updates the balance of each edge independently with constant probability $U(\mathbb{D}_{\mathbf{I}})$. Mathematically, for a path P with length L,

$$\mathbb{D}_{\mathbf{I}}[Q|P] = [U(\mathbb{D}_{\mathbf{I}})]^{|Q|}[1 - U(\mathbb{D}_{\mathbf{I}})]^{L-|Q|}, \quad \forall Q \subseteq P.$$

This noise mechanism enjoys similar privacy gains as the alternating mechanism due to path uncertainty. The main difference is that the number of candidate endpoints is now a random variable. We characterize the utility/privacy tradeoff precisely below.

Theorem 4.6. *Suppose $(\mathcal{V}, \mathcal{E})$ is a complete graph and $\mathcal{P}$ is the set of all simple paths of fixed length L. The privacy $\Pi(\mathbb{D}_{\mathbf{I}})$ of i.i.d. noise mechanism $\mathbb{D}_{\mathbf{I}}$ on a complete graph is lower bounded by*

$$\Pi(\mathbb{D}_{\mathbf{I}}) \geq 1 - \frac{2(1-\alpha)^L}{n} - \frac{2}{(L+1)(1-\alpha)}\left[1 - \alpha^{L+1} - (1-\alpha)^{L+1}\right], \tag{4.5}$$

where $\alpha = U(\mathbb{D}_{\mathbf{I}})$ is the utility.

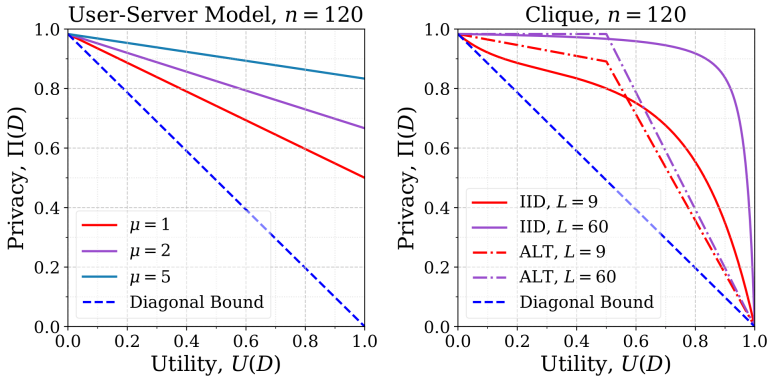

Fig. 7. Exploiting the structures of the networks, we can design mechanisms that overcome the diagonal bound of Theorem 3.2 (shown in blue dashed line). Under the user-server model, we exploit endpoint uncertainty and under the clique model, we exploit path uncertainty.

Figure 7 (right) plots the achievable bounds of Theorems 4.4 and 4.6, illustrating that neither scheme strictly dominates the other. However, i.i.d. noise has a larger jump in performance as we move from paths of length $L = 9$ to $L = 60$. Due to space constraints, the proofs of Theorems 4.4 and 4.6 can be found in Appendix A.

Takeaway Message. In this section, we show that the diagonal bound can be broken through two techniques: adding endpoint uncertainty and adding path uncertainty. Our exploration of path uncertainty in particular is limited to complete graphs due to the computational complexity of evaluating privacy metric (2.1) on general graphs. Still, we conjecture that similar intuitions can be applied to more complex, structured graphs; for instance, we relied on both the symmetry and the well-connected nature of complete graphs in our analysis; some circulant graphs exhibit similar properties. Hence, a takeaway message is that if one has the ability to choose a network structure, noise mechanism, *and* routing policy, it is possible to achieve an improved privacy-utility tradeoff; although there is no central entity controlling the graph topology, in practice system designers can influence topology by implementing default peer selection rules in client software. However, these benefits come at the cost of higher transaction fees due to the choice of routing through longer paths.[1] On the other hand, the user-server model arises organically in current payment channels, and is compatible with cost-effective, shortest-path routing.

5 RELATION BETWEEN OUR UTILITY METRIC AND SUCCESS RATE

A natural question is how to relate these results to the more intuitive (but more complicated to compute) success rate metric in Section 2. In this section, we explore the relation between our utility metric and success rate and observe that under some conditions, they are positively correlated.

Upon initial study, these two metrics actually appear to be *uncorrelated* due to an interesting phenomenon we refer to as "deadlocks." A channel is deadlocked when the public and true balances concentrate at different ends of the channel such that it cannot support any more transactions. For example, as shown in Figure 8, let Alice and Bob maintain a channel with capacity 2, where Alice holds 2 tokens and Bob holds 0. If the public balance shows the opposite, i.e., Bob claims to

[1]In general, a shorter path can cost more than a longer path because routers choose their own routing fees. Our model ignores this distinction for simplicity and because on average, longer paths tend to cost more.

Fig. 8. A deadlocked channel. No one else can send tokens through it in either direction.

hold 2 tokens and Alice 0, routers will try to send money only from Bob to Alice; this is impossible because Bob actually has no balance. Since no transactions pass through the channel, it will remain deadlocked forever, reducing the success rate over this channel to zero.

To demonstrate the deadlock effect, we show simulation results in the simplest setting, where the PCN consists of only 2 nodes A, B and 1 channel. The channel has capacity 10 and initial balance 5 on both ends. Each transaction, denoted by [value(sender $\rightarrow$ recipient)], is sampled from a uniform distribution over $\{2(A \rightarrow B), 2(B \rightarrow A), 3(A \rightarrow B), 3(B \rightarrow A)\}$. The sender is assumed not to know the true channel balance. Figure 9 plots the true balances vs the observed ones for 200 such independent channels as the number of transactions T grows; a deadlocked channel will show balances at the top-left and the bottom-right corners. Note that even at a high utility 0.9, almost all of the parallel single-edge PCNs are deadlocked after 100,000 transactions. In contrast, at perfect privacy (utility 0.0), no deadlock occurs because public channel balances remain forever at their initial value, $(5, 5)$.

Unfortunately, naive noise implementations can lead to severely deadlocked networks. To formally quantify this effect, consider the following model: assume that each transaction (i.e., its sender, recipient, and value) is independently sampled from some probability distribution. We fix a specific routing scheme, e.g., randomized shortest path routing. Under these assumptions, the process of transactions flowing through a PCN is a Markov chain. Each assignment of public and true balances is a state, and each transaction triggers a state transition. In the theorem below, we show that the steady-state success rate of this Markov chain is zero due to deadlocks. Intuitively, a sufficient number of channels will almost surely be eventually deadlocked after a finite number of transactions. These deadlocks block the entire PCN, resulting in a zero success rate.

Theorem 5.1 (deadlock). *Suppose a PCN $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is given with arbitrary topology and non-negative integer balances. Each transaction is independently sampled from an arbitrary distribution with value at least $\ell \in \mathbb{Z}_{++}$. The sender and recipient are selected symmetrically, and neither of them knows the true balance of any channel. Assume a noise mechanism $\mathbb{D}$ is applied to the transaction path, where the utility $\alpha \in (0, 1)$, and there exists an edge on the path, whose public balance is not updated with probability at least $\beta > 0$. Then, if the number of transactions $T \rightarrow \infty$, the PCN's success rate is* 0.

(Proof in Appendix A.6) Theorem 5.1 holds because long-term, *every* channel will become deadlocked and hence unusable. In particular, this implies that success rate is uncorrelated with utility. However, in practice, it is not true that route sources are unaware of balances of neighboring channels, or that there are infinitely many transactions. Although the theorem no longer applies without these assumptions, its insight regarding deadlocks helps to explain the counterintuitive non-monotonicity of the utility-success rate curves in Figures 10 and 11: one would intuitively expect success rate to increase when routers have access to more information. When utility is close but not equal to 0, the first edges on routes are likely to turn into deadlocks, negatively impacting success rate. If senders are aware of balances of neighboring channels, deadlocked channels can become unlocked by being the first edge on a route. This prevents success rates from falling to 0, but it is too small an effect to completely cancel the influence of deadlocks.

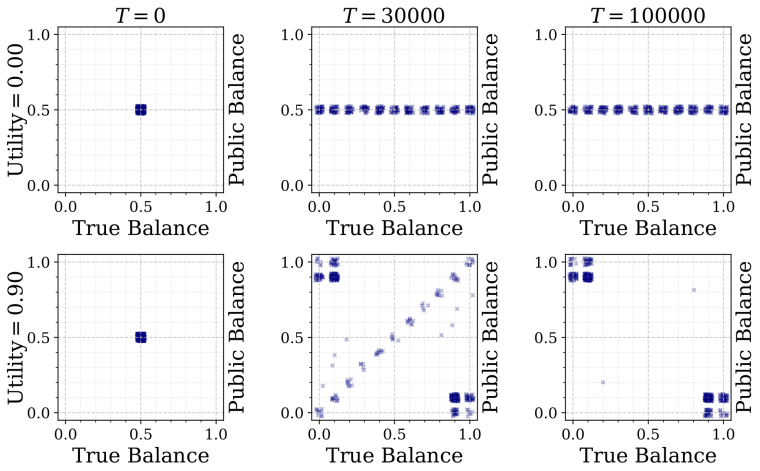

Fig. 9. Scatter plot of true balance vs public balance on both ends of all channels in 200 parallel experiments. Each point is plotted with a small deviation to demonstrate the number of overlapping points.

We consider two heuristic methods for alleviating deadlocks:
(1) Periodic Rebalancing. After every m global transactions (i.e., counted across the whole PCN), each deadlocked channel resets its *public* balances evenly to $(\frac{C}{2}, \frac{C}{2})$, where C denotes channel capacity, with probability 0.5. Otherwise, it remains deadlocked.
(2) Zero-valued Transactions. Introduce a new random process (e.g., Poisson) of 0-valued transactions, whose endpoints are selected uniformly from the participating nodes. These 0-transactions are subject to the same noise mechanism as regular transactions, and therefore update the public balances with some probability. If the public balance on an edge is not updated to the true balance, that edge's balance is instead reset to $(\frac{C}{2}, \frac{C}{2})$.

Notice that neither of these heuristics affects the privacy guarantee or the utility of the noise mechanism. Our privacy metric assumes a worst-case setting, where the initial public balances before a transaction are the real balances, and our utility metric considers the public balance updates from a single transaction. Hence, all the results from Sections 3 and 4 still apply. Nonetheless, we are not suggesting these countermeasures are ready to be put to use. For example, Periodic Rebalancing is difficult to deploy because in a decentralized distributed system, the nodes cannot easily know how many transactions happened globally. But most importantly, both heuristics leak balance information, introducing privacy drawbacks not captured by our metric. We leave analysis of these drawbacks to future work.

Now, we wish to understand the effect of these heuristics on success rate for a given utility and noise mechanism. Characterizing this theoretically is challenging, even for simple graphs like complete graphs; as such, we turn to simulations, shown in Figure 10. Here, utility represents the utility metric of all-or-nothing noise mechanism applied to the PCNs, which is defined in Section 3. Success rate is estimated by the portion of successful transactions over the total number, 100,000. At each utility point, these transactions are run on 50 parallel PCNs. Each channel is initialized with 1 token on both sides, and each transaction is of value 1. We test 3 different random topologies: 1) Erdős-Rényi networks with 50 nodes and expected density $\log 50/50$; 2) Erdős-Rényi networks with 50 nodes and expected density 0.25; and 3) preferential attachment networks (Barabási-Albert model). Initialized as a triangle, it has 47 new nodes inserted sequentially. Upon each insertion, the

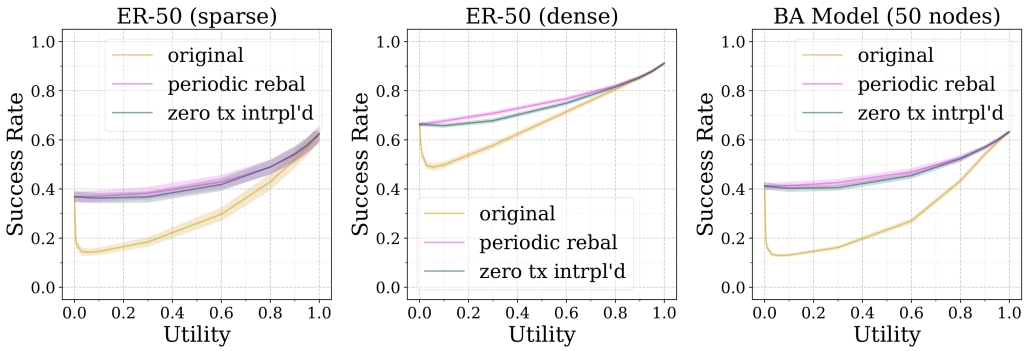

Fig. 10. Relations between our utility metric and success rate with and without deadlocks.

new node is attached to 2 existing nodes with probability proportional to the degree of existing node. For simplicity, we denote the graph as $\mathrm{BA}(K_3, 47, 2)$, where K_n denotes clique with n nodes. Note that in these simulations, even without alleviating deadlocks, the success rate is not exactly zero. This is partially because our simulations are limited in duration, and partially because nodes are assumed to know the balances on their adjacent channels. Both of these properties violate the assumptions in Theorem 5.1, but more realistically model a real PCN.

The solid curves and their error bars represent the means and the standard deviations of estimated success rates over 50 parallel PCNs, respectively. The length of each error bar equals twice the corresponding standard deviation. Figure 10 shows that both methods successfully alleviate the effects of deadlocks, eliminating the drop in success rate due to deadlocks. After this correction, our utility metric is monotonic in the success rate. Although this does not theoretically prove a monotonic relation between utility and success rate in general, it does suggest that optimizing the utility-privacy tradeoff as in Sections 3 and 4 is implicitly optimizing the relation between success rate and privacy in practice. We explore this relation further in the next section.

6 SIMULATIONS

In the previous section, we have shown empirically that utility and success rate are monotonically related when deadlocks are removed. We use this section to empirically investigate the tradeoff between success rate and privacy in simulation. The purpose of these simulations is twofold: first, we want to understand more carefully how the privacy-utility tradeoffs analyzed earlier translate into a privacy-success rate tradeoff. Second, we want to understand the effects of network properties not captured by our theoretical analysis, such as graph topology, transaction workload, and initial capacity distribution.

Our simulator models the sequential processing of a *workload*: a sequence of *transactions*, or tuples consisting of a sender, a receiver, and a transaction value. To match behavior in real deployments, each transaction is routed using shortest-path routing on the weighted graph, where routes are determined from public balances. A transaction fails if either no route can be found, or if the user tries a route whose true balances are insufficient to support the transaction. When a transaction fails, we do not allow retries. In line with theorems 3.2 and 3.6, we use the all-or-nothing noise scheme $\mathbb{D} = \mathbb{D}_{\mathbf{N}}$ with optimal privacy-utility tradeoff (3.3). Hence, after each transaction, we update all path balances with probability $U(\mathbb{D}_{\mathbf{N}})$, drawn independently for each transaction. We compute success rate by counting the fraction of successfully-processed transactions. Our theoretical results suggest that one can trade some privacy for an equal amount of utility. The question is, when we

sacrifice one unit of privacy for one of utility, does this translate into a gain of more than one unit of success rate?

Privacy-success rate curves depend on many factors, including the choice of network, workload, noise mechanism, and initial balance/capacity allocations. For each parameter setting, we generated 50 sets of networks and/or workloads in advance and ran each workload on its corresponding network, with the appropriate noise mechanism. We plot the mean success rate and standard error bars.

We looked into the effects of network structure, transaction value, workload size and network size on performance. We focus on network structure in this section and defer the other factors to Appendix B. We consider both synthetic and real network structures. For real network topologies, we use a snapshot of the **Lightning Network** and its channel capacities measured on December 28, 2018. Its major connected component consists of 2,266 nodes and 15,392 channels, while channel capacities vary from 1,100 to 16,777,216= 2^{24} satoshis (1 satoshi = 10^{-8} Bitcoin). For synthetic structures, we use four different options with equal number of nodes and expected number of edges to LND: 1) Erdős-Rényi graph with expected density 6.00×10^{-3}. 2) BA(BA(K_3, 113, 3), 2150, 7). Recall that BA(`init`, n, m) denotes a random graph initialized as `init`, to which n new nodes are added sequentially. On addition of each node, m new links to existing nodes are created. 3) A star-like network based on the user-server model where the server network is BA(K_{140}, 26, 137), and each user is attached to a server with probability proportional to server's degree. 4) An LND-like network. It is initialized as BA(BA(K_8, 75, 5), 965, 13), where degrees of nodes are relatively high (≥ 5). Then, each low-degree node is inserted and attached to 1, 2, 3 or 4 high-degree nodes uniformly at random. Eventually, it has equal number of nodes with degrees 1, 2, 3 and 4 to LND. This network is chosen to support our hypothesis about connection between success rate and number of low-degree nodes. We also explore the effects of varying network size in Appendix B.

In order to inspect the effects of topology, we set the capacity of every channel to be 1,000, and uniformly allocated balances on both ends of each channel. For the transaction workload, there is no canonical dataset available, so we uniformly and independently drew 2 nodes as the sender and receiver of each transaction, with transaction values independently following a Pareto distribution $\Psi(1.16, 1000)$. We chose a Pareto distribution to make the transaction value distribution heavy-tailed, and selected parameters corresponding to the Pareto principle. $\Psi(\beta, v_m)$ has CDF

$$F(v) = \left[1 - \left[\frac{v_m(\beta-1)}{\beta v}\right]^{\beta}\right] \cdot \mathbb{I}\left[v \geq \frac{v_m(\beta-1)}{\beta}\right].$$

Random variable $V \sim \Psi(\beta, v_m)$ has mean $\mathbb{E}[V] = v_m$. So our expected transaction value and channel capacities equal 1,000 sat.

The privacy-success rate tradeoff curves with error bars are shown in Figure 11. The width of error bars suggests a low standard error. Notice three trends:

Our first observation is the main takeaway message of this plot: **given a privacy not close to 1, we have the success rate hierarchy User-Server < LND ≈ LND-like < ER ≈ BA, coinciding with the reverse order of numbers of leaf nodes, which suggests a strong negative correlation between them.** This makes intuitive sense if we look into a specific leaf node in a PCN. Let privacy be 0 (i.e., noise is not applied), and Alice be a leaf node in the PCN, who has no tokens left on her only channel. We only consider transactions involving her, i.e., which she sends, receives, or relays. As a leaf node, she can't relay any transaction. Because she has no tokens, all of her attempts to send tokens out will fail, unless another node sends tokens back to her, rebalancing her channel. If she had a second channel, she would have more chances to successfully send tokens out. In addition, she would be able to relay transactions, which makes more flexible her connectivity

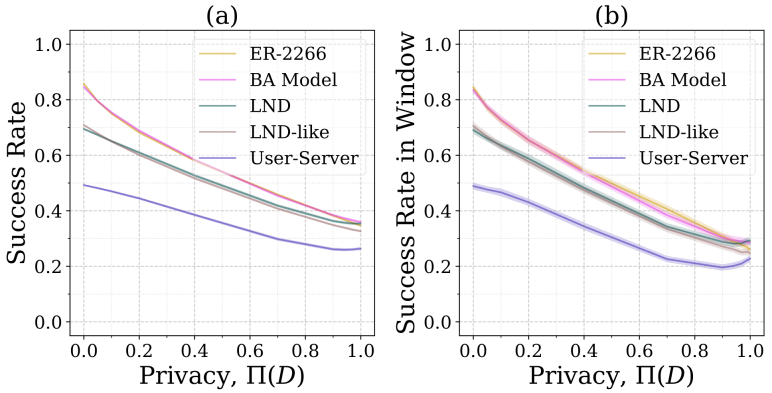

Fig. 11. Success rate-privacy curves on different network topologies. Success rate is the fraction of successful transactions out of total 100,000, while "success rate in window" is the fraction out of 2,000 most recent transactions.

to other nodes. This trend seems to remain when utility is not perfect, except when it's close to 0 and deadlocks emerge with higher probability. Noticing the user-server model has the lowest success rate curve, it is necessary for system designers to think of an addition trade-off between a gain in privacy (as proved in Section 4.1) and a loss in success rate, if choice of topology is available.

Second, "success rate in window" on Figure 11(b) has only a slight difference from success rate on Figure 11(a). Since the PCN is modelled as a Markov chain, both success rates will equal the probability of success of a next random transaction, when the Markov chain reaches its stable state. Naturally, the "success rate in window" taken from most recent 2,000 transactions could be an estimator of success rate at some point in the future. So we may predict that the non-monotonicity on Figure 11(b) will eventually appear on the success rate-privacy figure on Figure 11(a).

Third, on both sub-figures of Figure 11, if privacy is far enough away from 1, success rate decreases as privacy increases; but on Figure 11(b), when privacy approaches 1, success rate may increase along with privacy. This observation remains consistent with "original" curves in Figure 10. However, noticing the "increase" phase is much less significant than in Figure 10, we may assume deadlocks have much less effect on success rate of large-scale PCNs. Even without alleviation, a monotonic relation between success rate and utility (or privacy, of all-or-nothing noise) can be assumed.

Tradeoffs. Recall that one of our main goals was to understand if the one-to-one tradeoff of privacy for utility extends to success rate. Figure 11 suggests that this is not the case. For example, on the LND curve in Figure 11(a), if we start at today's operating point of perfect privacy, one must decrease privacy by 0.4 to gain 0.1 units of success rate. More generally, the curves in these figures are convex with low curvature; sacrificing some amount of privacy gives disproportionately small gains in success rate. This suggests that system designers must sacrifice significant amounts of privacy to achieve meaningful improvements in success rate. In addition, it's worth paying attention to the non-monotonicity of "success rate in window" in Figure 11(b). With a sufficiently large workload, one can always observe the decrease of success rate when privacy is sacrificed a little from perfect level. Hence system designers should be wary of deploying a noise mechanism that sacrifices only a little privacy; not only are the gains modest in theory, they can actually be negative in practice, unless system designers deal with deadlocks.

7 RELATED WORK

At a high level, our problem resembles privacy-preserving routing problems that have received substantial attention over the years [6, 7, 9, 12, 14, 15, 17, 19, 26, 29, 36]; however, PCNs introduce problem constraints that preclude the use of prior algorithms and analysis. For instance, early systems for receiver-anonymous messaging transmitted messages to additional nodes to hide the true destination [7, 19, 31]. This approach does not work in PCNs because messages are literally money, so sending a payment to additional nodes multiplies the cost of the transaction (the additional routing fees incurred by using longer paths are generally small in comparison). In other work [12, 36], a message is routed to prevent an adversarial observer from guessing the intended receiver prior to the message's delivery. This problem formulation is incompatible with PCNs, as all channel balances on the path are simultaneously updated only after a transaction has cleared. Moreover, the main concern in PCNs is that an adversary can infer the endpoints of a path *after* execution, not in an online setting. Finally, papers on source hiding in one-to-many broadcast settings do not apply to the point-to-point routing of PCNs, which updates balances only on a single path between two users [7, 14, 15].

Another relevant line of work analyzed the robustness of communication networks to adversarially-chosen corrupted links [8, 20, 21]. Roughly, these formulations can be interpreted as computing the worst-case success rate of a PCN following one or more adversarially-chosen transactions (though they do not consider privacy at all). A key difference is that in [8, 20, 21], the adversary directly corrupts a single edge at a time, whereas in a PCN, a transaction alters *all* the link balances in its path, and successive transactions can amplify imbalance. This prevents the analysis in [20, 21] from applying directly. However, this line of work suggests an interesting question: if one were to add a single channel to a PCN to maximize throughput under worst-case workloads, what channel should one add? This question is left for future work.

In parallel, PCNs have been increasingly studied by the academic community, typically focusing either on privacy or on utility. To our knowledge, our work is the first to explicitly quantify tradeoffs between privacy and utility in PCNs.

On the utility side, papers have primarily focused on improving the success rates of PCNs. This is typically achieved through new routing mechanisms, which exploit ideas from the networking literature, such as packet-switched routing, congestion control, and/or flow control [10, 32, 33]. Our work differs in assuming source-routed transactions, both for analytical tractability and because (almost) all existing implementations of PCNs today use source routing [2, 28]. Other recent utility work has instead focused on related but orthogonal issues, such as ensuring liveness in PCNs [25, 38] and rebalancing depleted channels [23].

On the privacy front, several papers have explored mechanisms for privacy-preserving PCN transaction routing [24, 30]. There are two key differences between these papers and our work. First, our adversary is a passive observer of the information publicly released to all participants, whereas prior work like [9, 17, 24, 30] considers a corrupt relay node that is trying to learn the destination of a transaction. Malavolta *et al.* [25] studied *passive value privacy*, or hiding the amount of a transaction to a passive external observer; they also study *active relationship anonymity*, which aims to hide the participants of a transaction to an active router. Our model is a combination of these; it considers a passive observer that wants to infer the endpoints of a transaction. Hence, our adversarial model is weaker than prior work in the sense that it sees less information than a router node, but stronger in the sense that encrypting the destination of a transaction does not solve the problem. Our results are already negative, so with a stronger active adversary, even worse tradeoffs could emerge. Second, our privacy problem is fundamentally different from that addressed in prior work. Because all network participants must be able to route transactions, encrypting transaction

balances is not a viable solution; statistical noise is therefore more natural. The most closely related work to ours ([25]) explicitly does not consider the problem of path selection, focusing instead on how to execute transactions once a path is selected.

Notice that differential privacy (a common statistical privacy metric) is difficult to implement in our setting [11]. For example, consider providing ε-local differential privacy on the source of a transaction [22]. This would require that for any observed path trace Q and any pair of candidate source nodes $u, v \in \mathcal{V}$, it must hold that $\mathbb{P}[Q|s(P) = v] \leq e^{\varepsilon}\mathbb{P}[Q|s(P) = u]$. To achieve this condition, we would need to add noise to edges that are not involved in the transaction itself; this adds significant overhead, both in terms of implementation and in terms of added noise. Like differential privacy, our metric is worst-case; however, it does not require hiding the transaction participants among all nodes in the graph.

8 DISCUSSION

In this paper, we have shown that for an important class of routing mechanisms (shortest-path routing), it is not possible to obtain a good privacy-utility tradeoff by releasing noisy channel balances in PCNs. The diagonal bound places an upper limit on the privacy-utility tradeoff obtainable in practice, and it is tight for general graph topologies. Our theoretical results are backed by simulations, which show that even if one considers a more complex end utility metric (i.e. success rate), the tradeoff does not improve; in fact, it appears to worsen in some cases. For example, we observe a new deadlocking phenomenon, by which channels end up severely imbalanced in one direction, while the public balances are severely imbalanced in the other. These deadlocks are impossible to revert for a transaction distribution with a minimum transaction value, such as the Pareto distributions used in our simulations. In practice, the situation may not be so dire; small transactions may be able to prevent the formation of total deadlocks. Another key observation is that our privacy analysis considers a passive adversary. An active adversary, which is stronger, could cause these tradeoffs to degrade.

Our findings suggest that network operators may not want to introduce noisy balance reporting mechanisms to trade off privacy for utility. Starting from today's operating point of perfect privacy, a network would need to give up almost all privacy to obtain substantial gains in success rate. As long as users use shortest-path routing, network operators may be better off choosing one extreme operating point or the other: perfect utility or perfect privacy.

Despite these pessimistic conclusions, our analysis does not close the book on this issue. It may be possible to obtain a more promising tradeoff by modeling the effects of concurrent transactions and packet-switched routing, where transactions are split into smaller units and sent over different paths. Packet-switched routing can cause transactions to flow over a much larger fraction of network edges per transaction. This may have a similar effect to the long-path analysis in Section 4.2 of confusing the adversary regarding the endpoints of the transaction. Analyzing such routing mechanisms is an important and interesting question for future work.

ACKNOWLEDGMENTS

This work is supported in part by NSF grants CIF-1705007 and CCF-1927712, ARO grant W911NF-17-S-0002, Input Output Hong Kong, the Franz Family Fund, and IC3. The authors would like to thank Kuang Xu for shepherding our paper, as well as the anonymous reviewers for their thoughtful comments.

REFERENCES

[1] Lightning network daemon. https://github.com/lightningnetwork/lnd.
[2] Raiden network. https://raiden.network/.

[3] Alessandro Acquisti, Leslie K John, and George Loewenstein. What is privacy worth? *The Journal of Legal Studies*, 42(2):249–274, 2013.

[4] Billy Bambrough. Bitcoin just crossed a huge adoption milestone. *Forbes*, 2019.

[5] Alastair R Beresford, Dorothea Kübler, and Sören Preibusch. Unwillingness to pay for privacy: A field experiment. *Economics letters*, 117(1):25–27, 2012.

[6] Srdjan Capkun, Jean-Pierre Hubaux, and Markus Jakobsson. Secure and privacy-preserving communication in hybrid ad hoc networks. Technical report, 2004.

[7] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

[8] William H Cunningham. Optimal attack and reinforcement of a network. *Journal of the ACM (JACM)*, 32(3):549–561, 1985.

[9] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.

[10] Mo Dong, Qingkai Liang, Xiaozhou Li, and Junda Liu. Celer network: Bring internet scale to every blockchain. *arXiv preprint arXiv:1810.00037*, 2018.

[11] Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[12] Mine Su Erturk and Kuang Xu. Dynamically protecting privacy, under uncertainty. *arXiv preprint arXiv:1911.08875*, 2019.

[13] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 45–59, 2016.

[14] Giulia Fanti, Peter Kairouz, Sewoong Oh, Kannan Ramchandran, and Pramod Viswanath. Hiding the rumor source. *IEEE Transactions on Information Theory*, 63(10):6679–6713, 2017.

[15] Giulia Fanti, Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Spy vs. spy: Rumor source obfuscation. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 271–284, 2015.

[16] David Floyd. Lightning network: What is it and can it solve bitcoin's scaling problem? *Investopedia*, March 2018.

[17] Michael J Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 193–206. ACM, 2002.

[18] Julio Gil-Pulgar. Atomic Multi-Path to help Bitcoin become a formidable payment instrument. *Bitcoinist*, February 2018.

[19] Sharad Goel, Mark Robson, Milo Polte, and Emin Sirer. Herbivore: A scalable and efficient protocol for anonymous communication. Technical report, Cornell University, 2003.

[20] Assane Gueye, Jean C Walrand, and Venkat Anantharam. Design of network topology in an adversarial environment. In *International Conference on Decision and Game Theory for Security*, pages 1–20. Springer, 2010.

[21] Assane Gueye, Jean C Walrand, and Venkat Anantharam. A network topology design game: How to choose communication links in an adversarial environment. In *Proc. of the 2nd international icst conference on game theory for networks, gamenets*, volume 11, page 5, 2011.

[22] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. In *Advances in neural information processing systems*, pages 2879–2887, 2014.

[23] Rami Khalil and Arthur Gervais. Revive: Rebalancing off-blockchain payment networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 439–453. ACM, 2017.

[24] Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, and Matteo Maffei. Silentwhispers: Enforcing security and privacy in decentralized credit networks. In *NDSS*, 2017.

[25] Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Srivatsan Ravi. Concurrency and privacy with payment-channel networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 455–471. ACM, 2017.

[26] Animesh Nandi, Armen Aghasaryan, and Makram Bouzid. P3: A privacy preserving personalization middleware for recommendation-based services. In *Hot Topics in Privacy Enhancing Technologies Symposium*, 2011.

[27] Dmytro Piatkivskyi and Mariusz Nowostawski. Split payments in payment networks. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 67–75. Springer, 2018.

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

[29] Michael K Reiter and Aviel D Rubin. Anonymous web transactions with crowds. *Communications of the ACM*, 42(2):32–48, 1999.

[30] Stefanie Roos, Pedro Moreno-Sanchez, Aniket Kate, and Ian Goldberg. Settling payments fast and private: Efficient decentralized routing for path-based transactions. *arXiv preprint arXiv:1709.05748*, 2017.

[31] Rob Sherwood, Bobby Bhattacharjee, and Aravind Srinivasan. P5: A protocol for scalable anonymous communication. *Journal of Computer Security*, 13(6):839–876, 2005.

[32] Vibhaalakshmi Sivaraman, Shaileshh Bojja Venkatakrishnan, Mohammad Alizadeh, Giulia Fanti, and Pramod Viswanath. Routing cryptocurrency with the spider network. In *HotNets*, 2019.
[33] Vibhaalakshmi Sivaraman, Shaileshh Bojja Venkatakrishnan, Kathleen Ruan, Parimarjan Negi, Lei Yang, Radhika Mittal, Giulia Fanti, and Mohammad Alizadeh. High throughput cryptocurrency routing in payment channel networks. In *NSDI*, 2020.
[34] Kyle Torpey. Greg maxwell: Lightning network better than sidechains for scaling bitcoin. *Bitcoin Magazine*, April 2016.
[35] Manny Trillo. Stress test prepares visanet for the most wonderful time of the year. http:stress-test-prepares-visanet-for-the-most-wonderful-time-of-the-year///www.visa.com/blogarchives/us/2013/10/10/index.html, 2013.
[36] John N Tsitsiklis and Kuang Xu. Delay-predictability trade-offs in reaching a secret goal. *Operations Research*, 66(2):587–596, 2018.
[37] Neel Varshney. Lightning network has 1 percent success rate with transactions larger than $200, controversial research says. *Hard Fork*, June 2018.
[38] Shira Werman and Aviv Zohar. Avoiding deadlocks in payment channel networks. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 175–187. Springer, 2018.
[39] Haifeng Yu, Ivica Nikolic, Ruomu Hou, and Prateek Saxena. Ohie: Blockchain scaling made simple. *arXiv preprint arXiv:1811.12628*, 2018.

APPENDIX

A PROOFS

We include proofs of the main results in this section.

A.1 Proof of Theorem 3.2

First, given that all routes are shortest paths, we prove the claim that there exists a unique pair of source and destination nodes of any path trace $Q \subseteq \mathcal{E}$. Let $\delta(u, v)$ denote the distance between any pair of nodes u, v on the graph. Assume by contradiction that Q is a path trace from x to y, and from z to w, where $\{x, y\} \neq \{z, w\}$. By definition, there exist paths P_{zw} and P_{xy}, where

(1) The endpoints of P_{zw} are z and w;
(2) The endpoints of P_{xy} are x and y;
(3) Q is a subset of both P_{zw} and P_{xy}.

As a result, both x and y are on path P_{zw}. Because P_{zw} is the shortest path connecting z and w, its segment connecting x and y is also the shortest path connecting x and y. Since $\{x, y\}$ is different from $\{z, w\}$, we have $\delta(x, y) < \delta(z, w)$. By the same reasoning, we get $\delta(z, w) < \delta(x, y)$ from z, w belonging to path P_{xy}, which is a contradiction. Therefore, any path trace $Q \subseteq \mathcal{E}$ with a source has a unique source. So there exists a function $s : 2^{\mathcal{E}} - \{\varnothing\} \to \mathcal{V}$, mapping each path trace to its source. Define $s(\varnothing) \notin \mathcal{V}$.

To upper bound our privacy metric, we will analyze the privacy achievable by an adversary with strategy $\mathbb{A}_{\mathbf{N}}$, where

(1) If no edge is updated by the noise mechanism, the adversary guesses randomly, i.e. $\mathbb{A}_{\mathbf{N}}[v|\varnothing] = \frac{1}{n}$ for all $v \in \mathcal{V}$;
(2) If a set of edges $Q \neq \varnothing$ is updated, the adversary estimates its source, i.e. $\mathbb{A}_{\mathbf{N}}[s(Q)|Q] = 1$.

Let $\varepsilon_s(P)$ denote the first edge of path P. By our privacy definition,

$$\begin{aligned}
\Pi(\mathbb{D}) &= 1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \sum_{Q \subseteq P} \mathbb{D}\left[Q|P\right] \mathbb{A}\left[\partial P|Q\right] \\
&\leq 1 - \min_{P \in \mathcal{P}} \sum_{Q \subseteq P} \mathbb{D}\left[Q|P\right] \mathbb{A}_{\mathbf{N}}[\partial P|Q] \\
&= 1 - \min_{P \in \mathcal{P}} \left[\frac{2\mathbb{D}\left[\varnothing|P\right]}{n} + \sum_{Q: s(Q)=s(P), Q \subseteq P} \mathbb{D}\left[Q|P\right] \right], \qquad \text{(A.1)}
\end{aligned}$$

where the last equality comes from splitting the adversary's probability of success into random guessing if no edges are updated, plus the probability of the noise mechanism revealing a path trace that truthfully reveals the true source edge. Recall that $\mathbb{A}\left[\partial P|Q\right]$ denotes the adversary's probability of guessing one of the endpoints of P upon observing path trace Q.

Because $\mathbb{D}[\cdot|P]$ is a probability distribution, the following constraints hold.

$$\begin{aligned}
\mathbb{D}[\varnothing|P] + \sum_{Q: s(Q)=s(P), Q \subseteq P} \mathbb{D}[Q|P] &\leq 1; \\
\sum_{Q: s(Q)=s(P), Q \subseteq P} \mathbb{D}[Q|P] = \sum_{Q: Q \ni \varepsilon_s(P), Q \subseteq P} \mathbb{D}[Q|P] &\geq U(\mathbb{D}),
\end{aligned}$$

where the latter condition follows from the definition of utility. If we assume $n \geq 2$, it follows that for any P,

$$\frac{2\mathbb{D}\left[\varnothing|P\right]}{n} + \sum_{Q:s(Q)=s(P),\, Q\subseteq P} \mathbb{D}\left[Q|P\right] \geq \frac{2}{n}[1-U(\mathbb{D})] + U(\mathbb{D}).$$

By substituting it back to (A.1), we obtain

$$\Pi(\mathbb{D}) \leq 1 - U(\mathbb{D}) - \frac{2}{n}[1-U(\mathbb{D})],$$

which is equivalent to (3.1). □

A.2 Proof of Proposition 4.1

For path $P \in \mathcal{P}$, let $\bar{P}$ denote P excluding the user-server channels, if they exist. If the all-or-nothing noise $\mathbb{D}_{\mathbf{N}}$ selects "nothing", then the adversary sees no update. Otherwise, the adversary will see $\bar{P}$ instead of P in our original model analyzed in Section 3. By definition of privacy,

$$\begin{aligned}
\Pi(\mathbb{D}_{\mathbf{N}}) &= 1 - \sup_{\mathbb{A}} \min_{P} \left\{\mathbb{A}[\partial P|\bar{P}]\alpha + \mathbb{A}[\partial P|\varnothing]\,(1-\alpha)\right\} \\
&= 1 - \sup_{\mathbb{A}[\cdot|\bar{P}],\mathbb{A}[\cdot|\varnothing]} \min_{P} \left\{\mathbb{A}[\partial P|\bar{P}]\alpha + \mathbb{A}[\partial P|\varnothing]\,(1-\alpha)\right\} \\
&\overset{(*)}{\leq} 1 - \sup_{\mathbb{A}[\cdot|\bar{P}],\mathbb{A}[\cdot|\varnothing]} \left[\alpha \min_{P} \mathbb{A}[\partial P|\bar{P}] + (1-\alpha) \min_{P'} \mathbb{A}[\partial P'|\varnothing]\right] \\
&= 1 - \alpha \sup_{\mathbb{A}[\cdot|\bar{P}]} \min_{P} \mathbb{A}[\partial P|\bar{P}] - (1-\alpha) \sup_{\mathbb{A}[\cdot|\varnothing]} \min_{P'} \mathbb{A}[\partial P'|\varnothing],
\end{aligned}$$

where the last line follows because $\mathbb{A}[\cdot|P]$ and $\mathbb{A}[\cdot|\varnothing]$ are independent. Because the network is reachable, we may apply Lemma 3.5 and obtain

$$\sup_{\mathbb{A}} \min_{P'} \mathbb{A}[\partial P'|\varnothing] = \frac{2}{n_S + n_U},$$

where the adversary reaches the supremum by choosing each node with equal probability when it sees no balance update.

Now we compute $\sup_{\mathbb{A}} \min_{P} \mathbb{A}[\partial P|\bar{P}]$. Since $\bar{P}$ must be a path between server nodes, we let x, y denote them and define two sets

$$X \triangleq \{x\} \cup \{v \in \mathcal{V}_U | J(v) = x\}\,,\ Y \triangleq \{y\} \cup \{v \in \mathcal{V}_U | J(v) = y\}\,.$$

If $x = y$, then the transaction only goes through user-server channels and $\bar{P} = \varnothing$. The adversary will see no update regardless of utility α, so random guessing with equal probability is the best adversarial strategy. Now we assume $x \neq y$, and immediately $X \cap Y = \varnothing$. The two endpoints of P must lie in each of X and Y. Because $\mathbb{A}[\cdot|\bar{P}]$ is a probability distribution, for any $\mathbb{A}$ there exists $u \in X$ and $v \in Y$, where

$$\mathbb{A}[u|\bar{P}] + \mathbb{A}[v|\bar{P}] \leq \max\left\{\frac{1}{|X|},\ \frac{1}{|Y|}\right\}.$$

By the properties of a user-server model, there exists a path P_u containing $\bar{P}$, and connecting u and v. Hence, we have

$$\sup_{\mathbb{A}} \min_{P} \mathbb{A}[\partial P'|\bar{P}'] \leq \sup_{\mathbb{A}} \mathbb{A}[\partial P_u|\bar{P}] \leq \max\left\{\frac{1}{|X|},\ \frac{1}{|Y|}\right\} \leq \frac{1}{\mu+1}. \tag{A.2}$$

Recall that $\mathbb{A}_{\mathbf{S}}$ was previously defined as $\mathbb{A}_{\mathbf{S}}[v|Q] = 1/(n_S + n_U)$ for all $v \in \mathcal{V}$, if $Q = \varnothing$. We supplement this definition by defining it also in the case $Q \neq \varnothing$. Recall that all-or-nothing noise $\mathbb{D}_{\mathbf{N}}$

is being used, and Q must be a shortest path on the server network $(\mathcal{V}_S, \mathcal{E}_S)$ from x' to y'. Similar to X and Y, we use the following notations.

$$X' = \{x'\} \cup \{v \in \mathcal{V}_U | J(v) = x'\}, \ Y' = \{y'\} \cup \{v \in \mathcal{V}_U | J(v) = y'\}.$$

Assume Z' is the set with fewer elements between X' and Y'. The strategy $\mathbb{A}_{\mathbf{S}}$ is defined as followed.

$$\mathbb{A}_{\mathbf{S}}[z|Q] = \begin{cases} \frac{1}{|Z'|}, & \text{if } z \in Z', \\ 0, & \text{otherwise.} \end{cases}$$

It can be easily confirmed that $\mathbb{A}_{\mathbf{S}}$ can make every inequality in (A.2) tight, thus

$$\sup_{\mathbb{A}} \min_{P} \mathbb{A}[\partial P|\bar{P}] = \frac{1}{\mu + 1}.$$

By substituting $\mathbb{A} = \mathbb{A}_{\mathbf{S}}$ in both sides of (*), we also get an equality. Hence,

$$\Pi(\mathbb{D}_{\mathbf{N}}) = 1 - \frac{1}{\mu + 1}\alpha - (1 - \alpha)\frac{2}{n_S + n_U},$$

which proves (4.1). □

A.3 Proof of Proposition 4.2

For path $P \in \mathcal{P}$, let $\bar{P}$ denote P excluding the user-server channels, if they exist. Furthermore, let $\mathcal{Q} = \left\{\bar{P} \middle| P \in \mathcal{P}\right\}$. By definition of privacy,

$$\begin{aligned}
&\Pi(\mathbb{D}_{\mathbf{N}}) \\
&= 1 - \sup_{\mathbb{A}} \min_{P} \left\{\mathbb{A}[\partial P|\bar{P}]\alpha + \mathbb{A}[\partial P|\varnothing](1 - \alpha)\right\} \\
&= 1 - \sup_{\mathbb{A}[\cdot|\neg\varnothing], \mathbb{A}[\cdot|\varnothing]} \min_{P} \left\{\mathbb{A}[\partial P|\bar{P}]\alpha + \mathbb{A}[\partial P|\varnothing](1 - \alpha)\right\} \\
&\overset{(*)}{\leq} 1 - \sup_{\mathbb{A}[\cdot|\neg\varnothing], \mathbb{A}[\cdot|\varnothing]} \left[\alpha \min_{P} \mathbb{A}[\partial P|\bar{P}] + (1 - \alpha) \min_{P'} \mathbb{A}[\partial P'|\varnothing]\right] \\
&= 1 - \alpha \sup_{\mathbb{A}[\cdot|\neg\varnothing]} \min_{P} \mathbb{A}[\partial P|\bar{P}] - (1 - \alpha) \sup_{\mathbb{A}} \min_{P'} \mathbb{A}[\partial P'|\varnothing] \\
&= 1 - \alpha \sup_{\mathbb{A}[\cdot|\neg\varnothing]} \min_{P} \mathbb{A}[\partial P|\bar{P}] - \frac{2}{n}(1 - \alpha) \\
&= 1 - \frac{2}{n}(1 - \alpha) - \alpha \min_{Q \in \mathcal{Q}} \sup_{\mathbb{A}[\cdot|Q]} \min_{P:\bar{P}=Q} \mathbb{A}[\partial P|Q]. \qquad (**)
\end{aligned}$$

Note that by Lemma 3.5, $\sup_{\mathbb{A}} \min_{P'} \mathbb{A}[\partial P'|\varnothing] = 2/n$. Any adversarial strategy can achieve this supremum by guessing uniformly at random when seeing no updates.

For a server node v, we define its **cloud** C_v as below.

$$C_v = \{v\} \cup \{u \in \mathcal{V}_U | (u, v) \in \mathcal{E}_U\}.$$

Apparently, for all $v \in \mathcal{E}_S$, $|C_v| \geq \mu + 1$. Let $x, y \in \mathcal{V}_S$ where $x \neq y$. Since C_x and C_y are not disjoint anymore, we define

$$Z = C_x \cap C_y, \ X = C_x - Z, \ Y = C_y - Z.$$

Apparently, X, Y and Z are disjoint subsets with union $C_x \cup C_y$. In addition, $x \in X$, $y \in Y$. For now, we assume $|Z| \geq 2$, so that there exists a path whose both source and destination belong to Z.

Because the network is reachable, there exists a path Q on the server network $(\mathcal{V}_S, \mathcal{E}_S)$ connecting x and y. In addition, for any $x' \in C_x$ and $y' \in C_y$ where $x' \neq y'$, there exists a path P with endpoints x', y', and containing Q. In this case, we have

$$\min_P \mathbb{A}_{\mathbf{S}}[\partial P|Q] = \min\{p_X + p_Y, p_X + p_Z, p_Y + p_Z, 2p_Z\},$$

where for $W \in \{X, Y, Z\}$, $p_W \triangleq \min_{v \in W} \mathbb{A}_{\mathbf{S}}[v|Q]$. Now obviously, the best adversary $\mathbb{A}_{\mathbf{S}}$ satisfies for all $x' \in X,\ y' \in Y,\ z' \in Z$,

$$(\mathbb{A}_{\mathbf{S}}[x'|Q],\ \mathbb{A}_{\mathbf{S}}[y'|Q],\ \mathbb{A}_{\mathbf{S}}[z'|Q]) \equiv (p_X,\ p_Y, p_Z).$$

Hence, to obtain $\mathbb{A}_{\mathbf{S}}[\cdot|Q]$, we only need to determine the probabilities p_X, p_Y and p_Z by solving the following problem.

$$\begin{array}{ll} \text{maximize} & \min\{p_X + p_Y,\ p_Y + p_Z,\ p_Z + p_X,\ 2p_Z\} \\ \text{subject to} & |X|\,p_X + |Y|\,p_Y + |Z|\,p_Z = 1. \end{array} \tag{A.3}$$

In order to deal with the minimum in the objective function, we discuss 4 cases.

Case 1: $p_Z \leq p_X, p_Y$. The objective function equals $2p_Z$, and

$$2p_Z \leq \frac{2\,(|X|\,p_X + |Y|\,p_Y + |Z|\,p_Z)}{|X| + |Y| + |Z|} = \frac{2}{|X| + |Y| + |Z|}.$$

By taking $p_X = p_Y = p_Z = 2/(|X| + |Y| + |Z|)$, the upper bound is attained. So in this case the maximum equals $2/(|X| + |Y| + |Z|)$.

Case 2: $p_Z \geq p_X, p_Y$. The objective function equals $p_X + p_Y$.

- If $||X| - |Y|| \leq |Z|$, we let $t = (|Y| - |X| + |Z|)/(2\,|Z|)$, which lies in $[0, 1]$. Then,

$$\begin{aligned} 1 &= |X|\,p_X + |Y|\,p_Y + |Z|\,p_Z \\ &\geq (|X| + t\,|Z|)\,p_X + [|Y| + (1-t)\,|Z|]\,p_Y \\ &= \frac{|X| + |Y| + |Z|}{2}\,(p_X + p_Y). \end{aligned}$$

  So the maximum equals $2/(|X| + |Y| + |Z|)$, which is attained by taking $p_X = p_Y = p_Z = 2/(|X| + |Y| + |Z|)$.
- If $|Y| > |X| + |Z|$, we have

$$\begin{aligned} 1 &= |X|\,p_X + |Y|\,p_Y + |Z|\,p_Z \\ &\geq (|X| + |Z|)\,p_X + (|X| + |Z|)\,p_Y. \end{aligned}$$

  The maximum equals $1/(|X| + |Z|)$, which is attained by taking $p_Y = 0$ and $p_X = p_Z = 1/(|X| + |Z|)$.
- If $|X| > |Y| + |Z|$, analogously, the maximum equals $1/(|Y| + |Z|)$ with maximizers $p_X = 0$ and $p_Y = p_Z = 1/(|Y| + |Z|)$.

To summarize, the maximum equals

$$\max\left\{\frac{2}{|X| + |Y| + |Z|},\ \frac{1}{|X| + |Z|},\ \frac{1}{|Y| + |Z|}\right\}.$$

Case 3: $p_X \le p_Z \le p_Y$. The objective function equals $p_X + p_Z$.

- If $|X| > |Y| + |Z|$,

$$\begin{aligned} 1 &= |X|\,p_X + |Y|\,p_Y + |Z|\,p_Z \\ &\ge (|Y| + |Z|)\,p_X + |Y|\,p_Z + |Z|\,p_Z. \end{aligned}$$

  This indicates maximum of $p_X + p_Z$ equals $1/(|Y| + |Z|)$ by picking $p_X = 0$ and $p_Y = p_Z = 1/(|Y| + |Z|)$.
- If $|X| \le |Y| + |Z|$,

$$\begin{aligned} 1 &= |X|\,p_X + |Y|\,p_Y + |Z|\,p_Z \\ &\ge |X|\,p_X + (|Y| + |Z|)p_Z \\ &\ge \frac{|X| + |Y| + |Z|}{2}\,(p_X + p_Z)\,. \end{aligned}$$

  The last line is obtained by using Chebyshev's sum inequality, with conditions $|X| \le |Y| + |Z|$ and $p_X \le p_Z$. So the maximum of $p_X + p_Z$ equals $2/(|X| + |Y| + |Z|)$, which is attained at $p_X = p_Y = p_Z = 1/(|X| + |Y| + |Z|)$.

To summarize, the maximum equals

$$\max\left\{\frac{2}{|X| + |Y| + |Z|},\ \frac{1}{|Y| + |Z|}\right\}.$$

Case 4: $p_Y \le p_Z \le p_X$. Analogously, the maximum equals

$$\max\left\{\frac{2}{|X| + |Y| + |Z|},\ \frac{1}{|X| + |Z|}\right\}.$$

In general, the maximum may be simply expressed as below.

$$\sup_{\mathbb{A}[\cdot|Q]} \min_{P} \mathbb{A}[\partial P|Q] = \max\left\{\frac{2}{|X| + |Y| + |Z|}, \frac{1}{|X| + |Z|}, \frac{1}{|Y| + |Z|}\right\}. \tag{A.4}$$

To attain it, the optimal adversarial strategy $\mathbb{A}_{\mathbf{S}}$ satisfies

(1) if $||X| - |Y|| \le |Z|$, then for all $v \in X \cup Y \cup Z$, $\mathbb{A}_{\mathbf{S}}[v|Q] = 1/(|X| + |Y| + |Z|)$;
(2) if $|X| > |Y| + |Z|$, then for all $x' \in X$ and $v \in Y \cup Z$, $\mathbb{A}_{\mathbf{S}}[x'|Q] = 0$, $\mathbb{A}_{\mathbf{S}}[v|Q] = 1/(|Y| + |Z|)$;
(3) if $|Y| > |X| + |Z|$, then for all $y' \in Y$ and $v \in X \cup Z$, $\mathbb{A}_{\mathbf{S}}[y'|Q] = 0$, $\mathbb{A}_{\mathbf{S}}[v|Q] = 1/(|X| + |Z|)$.

It should be addressed that the above solution applies only when $|Z| \ge 2$. For the case $|Z| = 0$, we have $C_x \cap C_y = \varnothing$ equivalently. Proof of Proposition 4.1 has shown that the maximum value and the optimal strategy are actually compatible with (A.4). Hence, we only need to focus on the case $|Z| = 1$. The analysis begins to fork at the problem formulation, because there will be no path with both ends lying in set Z. So we need to modify (A.3) as followed.

$$\begin{aligned} \text{maximize} \quad & \min\{p_X + p_Y,\ p_Y + p_Z,\ p_Z + p_X\} \\ \text{subject to} \quad & |X|\,p_X + |Y|\,p_Y + |Z|\,p_Z = 1. \end{aligned} \tag{A.5}$$

After similar discussions, we can obtain the maximum

$$\max\left\{\frac{2}{|X| + |Y| + |Z|}, \frac{1}{|X| + |Z|}, \frac{1}{|Y| + |Z|}, \frac{1}{|X| + |Y|}\right\}.$$

Because $|Z| = 1 \le |X|, |Y|$, both (A.4) and $\mathbb{A}_{\mathbf{S}}$ apply to this case. So we may regard them as a general solution, without respect to the value of $|Z|$. Recalling μ is the minimum number of user

nodes connected to any server,

$$\begin{aligned}
&\max\left\{\frac{2}{|X|+|Y|+|Z|}, \frac{1}{|X|+|Z|}, \frac{1}{|Y|+|Z|}\right\} \\
=\ &\max\left\{\frac{2}{\left|C_x \cup C_y\right|}, \frac{1}{|C_x|}, \frac{1}{\left|C_y\right|}\right\} \\
\le\ &\max\left\{\frac{2}{\mu+2}, \frac{1}{\mu+1}, \frac{1}{\mu+1}\right\} = \frac{2}{\mu+2}.
\end{aligned}$$

Now we go back to privacy. It's easy to confirm that by substituting the supremum over $\mathbb{A}$ with $\mathbb{A}_{\mathbf{S}}$ at both sides of (*), it turns into an equality. Following (**),

$$\begin{aligned}
\Pi(\mathbb{D}_{\mathbf{N}}) &= 1 - \frac{2}{n}(1-\alpha) - \alpha \min_{Q\in\mathcal{Q}} \sup_{\mathbb{A}[\cdot|Q]} \min_{P:\bar{P}=Q} \mathbb{A}[\partial P|Q] \\
&\ge 1 - \frac{2}{n}(1-\alpha) - \alpha \min_{Q\in\mathcal{Q}} \frac{2}{\mu+2} \\
&= \left(1-\frac{2}{n}\right)(1-\alpha) + \frac{\mu}{\mu+2}\alpha.
\end{aligned}$$

So the lower bound of privacy is justified. □

A.4 Proof of Theorem 4.4

Let $\ell = \lfloor L/2 \rfloor$. Let $\bar{P}_1$ and $\bar{P}_2$ denote the "odd" group and "even" group of path P, respectively. We define

$$\begin{aligned}
\mathcal{Q}_i &\triangleq \left\{\bar{P}_i \middle| P \in \mathcal{P}\right\},\ i \in \{1,2\}, \\
Z(v,Q) &\triangleq \left|\left\{P \middle| v \in \partial P,\ Q \in \left\{\bar{P}_1, \bar{P}_2\right\}\right\}\right|.
\end{aligned}$$

The method to analyse $Z(v,Q)$ depends on the parity of L.

Case 1: When $2 \mid L$, we have $\mathcal{Q}_1 = \mathcal{Q}_2$ (because the graph is complete). For convenience and consistency, we introduce some terminology that will be reused in the proof of Theorem 4.6. For any path trace Q of P, it consists of several disconnected segments, where each segment consists of one or more edges connected end to end (in the alternating noise scheme, each segment is only one edge). Each segment has two end nodes and zero or more intermediate nodes. We call the end nodes of a segment **gray nodes**, and the intermediate nodes are called **white nodes**. Again, in the alternating noise scheme, there are no white nodes; we will use white nodes in the proof of Theorem 4.6. The rest of the nodes in the network are called **black nodes**. Expressing these as sets, we let g_Q, w_Q and b_Q denote the sets of gray, white and black nodes given path trace Q. An example of segments and the aforementioned colors are shown in Figure 12.

For any path P under the alternating noise scheme, both $\bar{P}_1$ and $\bar{P}_2$ yield $L = 2\ell$ gray nodes and 1 black node. All nodes not on path P are always black nodes. If a gray node v is determined to be one endpoint of P, then its status as a source or destination is determined too. In that case, the other end must be a black node. So $Z(v,Q)$ equals the number of ways to pick this black node, multiplied by the number of ways to permute the remaining $\ell - 1$ segments. If a black node v is determined to be one end, it could be either a source or a destination. Then, $Z(v,Q)$ equals the number of ways to permute all ℓ segments, multiplied by 2, the number of ways to determine which end v is. That is,

$$Z(v,Q) = \begin{cases} (\ell-1)!(n-2\ell), & v \in g_Q; \\ 2\ell!, & v \in b_Q. \end{cases}$$

Case 2: When $2 \nmid L$, the case is different because for all $Q \in \boldsymbol{Q}_1$, Q has $\ell + 1$ edges, while for all $Q \in \boldsymbol{Q}_2$, Q has ℓ edges. As a result, $\boldsymbol{Q}_1 \cap \boldsymbol{Q}_2 = \varnothing$. For all $Q \in \boldsymbol{Q}_1$,

$$Z(v, Q) = \begin{cases} \ell!, & v \in g_Q; \\ 0, & v \in b_Q. \end{cases}$$

For all $Q \in \boldsymbol{Q}_2$,

$$Z(v, Q) = \begin{cases} 0, & v \in g_Q; \\ 2\ell!(n - 2\ell - 1), & v \in b_Q. \end{cases}$$

Define adversarial strategy $\mathbb{A}_{\mathbf{T}}$ as below. Note that $\mathbb{A}_{\mathbf{T}}[v|\varnothing] = 1/n$ for all $v \in \mathcal{V}$. If balance updates of the entire path P are visible, $\mathbb{A}_{\mathbf{T}}$ directly picks an end node of it, so for all $P \in \mathcal{P}$, $\mathbb{A}_{\mathbf{T}}[\partial P|P] = 1$.

(1) When $2 \mid L$,

$$\mathbb{A}_{\mathbf{T}}[v|Q] = \begin{cases} \dfrac{\mathbb{I}[2L < n]}{L} + \dfrac{\mathbb{I}[2L = n]}{n}, & v \in g_Q; \\ \dfrac{\mathbb{I}[2L = n]}{n} + \dfrac{\mathbb{I}[2L > n]}{n - L}, & v \in b_Q. \end{cases}$$

(2) When $2 \nmid L$,

$$\mathbb{A}_{\mathbf{T}}[v|Q] = \begin{cases} \dfrac{\mathbb{I}[Q \in \boldsymbol{Q}_1]}{L + 1}, & v \in g_Q; \\ \dfrac{\mathbb{I}[Q \in \boldsymbol{Q}_2]}{n - L + 1}, & v \in b_Q. \end{cases}$$

Based on calculations of $Z(v, Q)$, it's easy to confirm that

$$\mathbb{A}_{\mathbf{T}}[\cdot|Q] \in \operatorname*{argsup}_{\mathbb{A}[\cdot|Q]} \sum_{v \in \mathcal{V}} \mathbb{A}[v|Q] Z(v, Q), \quad \forall Q \in \boldsymbol{Q}_1 \cup \boldsymbol{Q}_2.$$

Now we analyse privacy based on the following discussions.

(1) When $\alpha \le 1/2$,

$$\begin{aligned} & 1 - \Pi(\mathbb{D}_{\mathbf{T}}) \\ = & \sup_{\mathbb{A}} \inf_{P \in \mathcal{P}} \left\{ \alpha \mathbb{A}[\partial P|\bar{P}_1] + \alpha \mathbb{A}[\partial P|\bar{P}_2] + (1 - 2\alpha) \mathbb{A}[\partial P|\varnothing] \right\} \\ = & \alpha \sup_{\mathbb{A}} \inf_{P \in \mathcal{P}} \left\{ \mathbb{A}[\partial P|\bar{P}_1] + \mathbb{A}[\partial P|\bar{P}_2] \right\} + \frac{2 - 4\alpha}{n} \qquad \text{(A.6)} \\ \overset{(\dagger)}{\le} & \frac{\alpha}{|\mathcal{P}|} \sup_{\mathbb{A}} \sum_{P \in \mathcal{P}} \left\{ \mathbb{A}[\partial P|\bar{P}_1] + \mathbb{A}[\partial P|\bar{P}_2] \right\} + \frac{2 - 4\alpha}{n} \\ = & \frac{\alpha}{|\mathcal{P}|} \sup_{\mathbb{A}} \sum_{Q \in \boldsymbol{Q}_1 \cup \boldsymbol{Q}_2} \sum_{v \in \mathcal{V}} \mathbb{A}[v|Q] Z(v, Q) + \frac{2 - 4\alpha}{n}. \end{aligned}$$

Note that (A.6) is obtained by Lemma 3.5, where $\mathbb{A}_{\mathbf{T}}[\cdot|\varnothing]$ achieves the supremum. If we plug $\mathbb{A} = \mathbb{A}_{\mathbf{T}}$ into both sides of (†) by replacing the supremum, we get an equality. Hence, the left side supremum of (†) is also achieved by $\mathbb{A}_{\mathbf{T}}$. Therefore, we can calculate $\Pi(\mathbb{D}_{\mathbf{T}})$ as followed.

(a) When $2 \mid L$, let $P \in \mathcal{P}$ be arbitrary. In this case, both $\bar{P}_1$ and $\bar{P}_2$ have a gray end node and a black end node. Hence,

$$\begin{aligned} & \Pi(\mathbb{D}_{\mathbf{T}}) \\ = & 1 - \alpha \left(\mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_1] + \mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_2] \right) - \frac{2 - 4\alpha}{n} \end{aligned}$$

$$= 1 + \frac{4\alpha - 2}{n} - 2\alpha\left(\frac{\mathbb{I}[2L < n]}{L} + 2\frac{\mathbb{I}[2L = n]}{n} + \frac{\mathbb{I}[2L > n]}{n - L}\right)$$
$$= 1 - \frac{2}{n} - \left(\frac{2}{\min\{L, n - L\}} - \frac{4}{n}\right)\alpha.$$

(b) When $2 \nmid L$, let $P \in \mathcal{P}$ be arbitrary. Both end nodes are gray for $\bar{P}_1$ and black for $\bar{P}_2$. Hence,

$$\Pi(\mathbb{D}_{\mathbf{T}}) = 1 - \alpha\left(\mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_1] + \mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_2]\right) - \frac{2 - 4\alpha}{n}$$
$$= 1 + \frac{4\alpha - 2}{n} - \alpha\left(\frac{2\mathbb{I}[\bar{P}_1 \in \mathcal{Q}_1]}{L + 1} + \frac{2\mathbb{I}[\bar{P}_2 \in \mathcal{Q}_2]}{n - L + 1}\right)$$
$$= 1 - \frac{2}{n} - \left(\frac{2}{L + 1} + \frac{2}{n - L + 1} - \frac{4}{n}\right)\alpha.$$

(2) When $\alpha > 1/2$, with probability $(2\alpha - 1)$, public balances of P are updated. Hence,

$$\begin{aligned}
&\Pi(\mathbb{D}_{\mathbf{T}}) \\
&= 1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \left\{(1 - \alpha)\mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_1] + (1 - \alpha)\mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_2] + \right. \\
&\qquad\qquad \left. (2\alpha - 1)\mathbb{A}_{\mathbf{T}}[\partial P|P]\right\} \\
&= 1 - (2\alpha - 1) - (1 - \alpha)\sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \left\{\mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_1] + \mathbb{A}_{\mathbf{T}}[\partial P|\bar{P}_2]\right\}.
\end{aligned}$$

Note that the supremum has been solved in the case $\alpha \le 1/2$. So the results will be written below directly.

(a) When $2 \mid L$,

$$\Pi(\mathbb{D}_{\mathbf{T}}) = (1 - \alpha)\left(2 - \frac{2}{\min\{L, n - L\}}\right).$$

(b) When $2 \nmid L$,

$$\Pi(\mathbb{D}_{\mathbf{T}}) = (1 - \alpha)\left(2 - \frac{2}{L + 1} - \frac{2}{n - L + 1}\right).$$

The tradeoffs have been proved by the discussions above. □

A.5 Proof of Theorem 4.6

Let $\lambda = L + 1$ denote the number of nodes in the path. Define $\mathcal{Q} \triangleq \bigcup_{P \in \mathcal{P}} 2^P$, where 2^S denotes the class of subset of an arbitrary set S. Notice that the adversarial strategy $\mathbb{A}$ has only one constraint – for every $Q \in \mathcal{Q}$, $\mathbb{A}[\cdot|Q]$ is a probability distribution over $\mathcal{V}$, while $\mathbb{A}[\cdot|Q_1]$ is independent with $\mathbb{A}[\cdot|Q_2]$ for $Q_1 \neq Q_2$. Hence, we can divide $\mathbb{A}$ into many separate distributions with a small sample space $\mathcal{V}$.

Beginning with definition of privacy, we use a trick of comparing minimum and mean.

$$\begin{aligned}
\Pi(\mathbb{D}_{\mathbf{I}}) &= 1 - \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} \sum_{Q \subseteq P} \mathbb{A}[\partial P|Q]\mathbb{D}_{\mathbf{I}}[Q|P] \\
&\ge 1 - \sup_{\mathbb{A}} \frac{1}{|\mathcal{P}|} \sum_{P \in \mathcal{P}} \sum_{Q \subseteq P} \mathbb{A}[\partial P|Q]\mathbb{D}_{\mathbf{I}}[Q|P] \qquad (A.7) \\
&= 1 - \frac{1}{|\mathcal{P}|} \sup_{\mathbb{A}} \sum_{Q \in \mathcal{Q}} \alpha^{|Q|}(1 - \alpha)^{L - |Q|} \sum_{P \in \mathcal{P}, P \supset Q} \mathbb{A}[\partial P|Q]
\end{aligned}$$

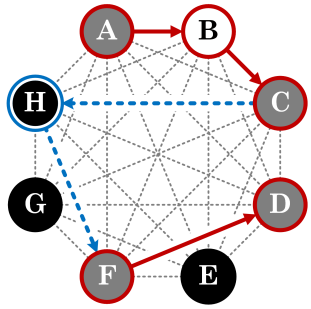

Fig. 12. A path and its path trace for i.i.d. noise. Gray nodes are the endpoints of a revealed segment; white nodes are internal to a revealed segment; black nodes are not adjacent to any revealed edge.

$$\triangleq 1 - \sum_{Q \in \mathcal{Q}} \frac{(n-\lambda)!\alpha^{|Q|}(1-\alpha)^{L-|Q|}}{n!} \cdot \sup_{\mathbb{A}[\cdot|Q]} \sum_{v \in \mathcal{V}} \mathbb{A}[v|Q] Z(v, Q). \tag{A.8}$$

(A.8) ended with substitution $Z(v, Q) = |\{P \in \mathcal{P} | Q \subseteq P, v \in \partial P\}|$. Since Q is a path trace of P, it consists of several disconnected segments, where each segment consists of one or more edges connected end to end. Intuitively, a segment has two end nodes and zero or more intermediate nodes. The end nodes of segments are called **gray nodes**, and the intermediate nodes are called **white nodes**. The rest of the nodes in the network are called **black nodes**. Expressing these as sets, we let g_Q, w_Q and b_Q denote the sets of gray, white and black nodes given path trace Q. An example of segments and the aforementioned colors are shown in Figure 12.

The colors of nodes are decided by Q consisting of disconnected segments. Assume Q has $h_Q \geq 1$ segments. Furthermore, let k_Q denote the number of *internal* black nodes, i.e. the black nodes *on the original path*. Relatively, *external* black nodes are the nodes out of the original path. Now the number of white nodes equals $\lambda - 2h_Q - k_Q$ and that of external black nodes equals $n - \lambda$. The value of $Z(v, Q)$ is discussed based on the color of v.

(1) If v is a white node, it could never be an end node of a path, so $Z(v, Q) = 0$ for all $v \in w_Q$.
(2) If v is a gray node, we first deal with the case that v is a source node of some segment σ_0, where the other segments are denoted by σ_1 through σ_{h_Q-1}. Each original path with source node v has a permutation of the rest L nodes, which is uniquely mapped to a combination of the following options.
   - Order of segments on the path. There are $(h_Q - 1)!$ ways to permute q_1 through q_{n-1}.
   - Distribution of black nodes in the gaps between consecutive segments, or in the extension of the last segment. There are $\binom{h_Q + k_Q - 1}{k_Q}$ different ways to distribute the slots for black nodes.
   - Permutation of black nodes filling into the slots. Without regard of the position of the slots, there are $(n - \lambda + k_Q)!/(n - \lambda)!$ different permutations.

   These options are independent of each other, so the multiplication rule applies. The analysis for v being a destination node of a segment is identical. It follows that for all $v \in g_Q$,

$$Z(v, Q) = \frac{(h_Q + k_Q - 1)!(n - \lambda + k_Q)!}{k_Q!(n - \lambda)!}. \tag{A.9}$$

(3) If v is a black node, we must assume $k_Q \geq 1$. We first deal with the case that v is a source node of the path. Similarly we discuss the following options.
- Order of segments. There are $h_Q!$ different orders.
- Distribution of the remaining $k_Q - 1$ slots for black nodes. There are $\binom{h_Q + k_Q - 1}{k_Q - 1}$ different distributions.
- Permutation of black nodes filling into the slots. There are $(n-\lambda+k_Q-1)!/(n-\lambda)!$ different permutations.

The analysis for v being a destination node is identical. Considering v could be both source nodes and a destination nodes of different paths, for all $v \in b_Q$,

$$Z(v,Q) = \begin{cases} \dfrac{2(h_Q + k_Q - 1)!(n - \lambda + k_Q - 1)!}{(k_Q - 1)!(n-\lambda)!}, & k_Q \geq 1; \\ 0, & k_Q = 0. \end{cases} \tag{A.10}$$

$Z(v,Q)$ depends only on the color of v, i.e. the set v belongs to. We may use $Z_g(Q)$ and $Z_b(Q)$ to denote $Z(v,Q)$ for an arbitrary $v \in g_Q$ and $v \in b_Q$, respectively. Starting with the LP in (A.8),

$$\begin{aligned} &\sup_{\mathbb{A}[\cdot|Q]} \sum_{v \in \mathcal{V}} \mathbb{A}[v|Q] Z(v,Q) \\ =& \sup_{\mathbb{A}[\cdot|Q]} \left[Z_g(Q) \sum_{u \in g_Q} \mathbb{A}[u|Q] + Z_b(Q) \sum_{v \in b_Q} \mathbb{A}[v|Q] \right] \\ =& \max\left\{ Z_g(Q),\ Z_b(Q) \right\} \\ =& \frac{(h_Q + k_Q - 1)!(n - \lambda + k_Q)!}{(n-\lambda)! k_Q!} \psi(k_Q), \end{aligned}$$

where

$$\psi(k_Q) = \begin{cases} 1, & k_Q \leq n - \lambda; \\ \frac{2k_Q}{n - \lambda + k_Q}, & \text{otherwise.} \end{cases}$$

One of the maximizers $\mathbb{A}_{\mathbf{I}}$ of the LP could be written as

$$\mathbb{A}_{\mathbf{I}}[v|Q] = \begin{cases} \dfrac{\mathbb{I}[v \in g_Q]}{2h_Q}, & k_Q \leq n - \lambda; \\ \dfrac{\mathbb{I}[v \in b_Q]}{n - \lambda + k_Q}, & \text{otherwise.} \end{cases}$$

Explicitly, when the adversary sees balance updates on path trace Q, it picks a color from gray and black, and randomly chooses a node of that color. The choice of color depends on whether $k_Q \leq n - \lambda$. Recall that k_Q is the number of internal black nodes, and $n - \lambda$ is the number of external black nodes.

Let $f(\mathbb{A}, P) \triangleq \sum_{Q \subseteq P} \mathbb{A}[\partial P|Q] \mathbb{D}_{\mathbf{I}}[Q|P]$. Based on the symmetry of $\mathbb{A}_{\mathbf{I}}$, it's easy to confirm that when substituting the supremum over $\mathbb{A}$ with $\mathbb{A}_{\mathbf{I}}$ in (A.7), it will make the inequality tight. Let $\mathbb{U}$ denote the uniform distribution over $\mathcal{P}$. Now we obtain

$$\begin{aligned} \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} f(\mathbb{A}, P) &\leq \sup_{\mathbb{A}} \mathop{\mathbb{E}}_{P \sim \mathbb{U}} f(\mathbb{A}, P) \\ &= \mathop{\mathbb{E}}_{P \sim \mathbb{U}} f(\mathbb{A}_{\mathbf{I}}, P) = \min_{P \in \mathcal{P}} f(\mathbb{A}_{\mathbf{I}}, P) \\ &\leq \sup_{\mathbb{A}} \min_{P \in \mathcal{P}} f(\mathbb{A}, P). \end{aligned}$$

Hence, (A.7) is actually an equality, and

$$\mathbb{A}_{\mathbf{I}} \in \operatorname*{argsup}_{\mathbb{A}} \min_{P\in\mathcal{P}} \sum_{Q\subseteq P} \mathbb{A}[\partial P|Q]\mathbb{D}_{\mathbf{I}}[Q|P].$$

This means $\mathbb{A}_{\mathbf{I}}$ is an optimal adversarial strategy for i.i.d. noise $\mathbb{D}_{\mathbf{I}}$ and current routing scheme. The proof will be wrapped up with a final computation of $\Pi(\mathbb{D}_{\mathbf{I}})$ based on the optimal $\mathbb{A}_{\mathbf{I}}$. Since $\min_{P\in\mathcal{P}} f(\mathbb{A}_{\mathbf{I}}, P) = \mathbb{E}_{P\sim\mathbb{U}} f(\mathbb{A}_{\mathbf{I}}, P)$, for an arbitrary $P' \in \mathcal{P}$,

$$\min_{P\in\mathcal{P}} f(\mathbb{A}_{\mathbf{I}}, P) = f(\mathbb{A}_{\mathbf{I}}, P').$$

Immediately,

$$\Pi(\mathbb{D}_{\mathbf{I}}) = 1 - \sum_{Q\subseteq P'} \alpha^{|Q|}(1-\alpha)^{L-|Q|}\mathbb{A}_{\mathbf{I}}[\partial P'|Q]. \tag{A.11}$$

Let v_s and v_d denote the source and destination of path P', respectively. For a particular Q with $h \geq 1$ segments and k internal black nodes, we have $|Q| = \lambda - h - k$. Now we discuss the colors of v_s and v_d, depending on Q.

(1) If both are black, then

$$\mathbb{A}_{\mathbf{I}}[v_s|Q] = \mathbb{A}_{\mathbf{I}}[v_d|Q] = \frac{\mathbb{I}[k > n-\lambda]}{n-\lambda+k}.$$

Because the path is given, the number of path traces with both black ends equals the number of arrangements of slots for black nodes and white nodes. In this case, there are $h+1$ gaps for $k-2$ black nodes, and h gaps for $\lambda - 2h - k$ white nodes. Hence,

$$\left|\left\{Q \subseteq P \middle| v_s, v_d \in b_Q\right\}\right| = \binom{h+k-2}{k-2}\binom{\lambda-h-k-1}{h-1}.$$

(2) If v_s is black and v_d is gray, then

$$\mathbb{A}_{\mathbf{I}}[v_s|Q] = \frac{\mathbb{I}[k > n-\lambda]}{n-\lambda+k},\ \mathbb{A}_{\mathbf{I}}[v_d|Q] = \frac{\mathbb{I}[k \leq n-\lambda]}{2h}.$$

Similarly, we find the number of path traces Q by finding the number of ways to put $k-1$ slots of black nodes into h gaps, and $\lambda - 2h - k$ slots of white nodes into h gaps. Hence,

$$\left|\left\{Q \subseteq P \middle| v_s \in b_Q, v_d \in g_Q\right\}\right| = \binom{h+k-2}{k-1}\binom{\lambda-h-k-1}{h-1}.$$

(3) If v_s is gray and v_d is black, by symmetry,

$$\mathbb{A}_{\mathbf{I}}[v_s|Q] = \frac{\mathbb{I}[k \leq n-\lambda]}{2h},\ \mathbb{A}_{\mathbf{I}}[v_d|Q] = \frac{\mathbb{I}[k > n-\lambda]}{n-\lambda+k},$$

$$\left|\left\{Q \subseteq P \middle| v_s \in g_Q, v_d \in b_Q\right\}\right| = \binom{h+k-2}{k-1}\binom{\lambda-h-k-1}{h-1}.$$

(4) If both are gray, then

$$\mathbb{A}_{\mathbf{I}}[v_s|Q] = \mathbb{A}_{\mathbf{I}}[v_d|Q] = \frac{\mathbb{I}[k \leq n-\lambda]}{2h},$$

$$\left|\left\{Q \subseteq P \middle| v_s \in g_Q, v_d \in b_Q\right\}\right| = \binom{h+k-2}{k}\binom{\lambda-h-k-1}{h-1}.$$

The case $k < 2$ is not specified as $\binom{M}{N} = 0$ when $M < 0 \leq N$.

After the above discussion, we may conclude that for a particular Q, its term in the summation depends only on its h, k and the color of v_S and v_d. Therefore, the 2^λ terms in the original summation (A.11) are merged into Poly(n) terms as below.

$$
\begin{aligned}
& 1 - \Pi(\mathbb{D}_{\mathbf{I}}) - \frac{2(1-\alpha)^L}{n} \\
= & \sum_{h=1}^{\lfloor \lambda/2 \rfloor} \sum_{k=0}^{\lambda-2h} \alpha^{\lambda-h-k}(1-\alpha)^{h+k-1} \cdot \binom{\lambda-h-k-1}{h-1} \cdot 2 \cdot \\
& \quad \left\{ \binom{h+k-2}{k-2} \frac{\mathbb{I}[k > n-\lambda]}{n-\lambda+k} \right. \\
& \quad + \binom{h+k-2}{k-1} \left[\frac{\mathbb{I}[k > n-\lambda]}{n-\lambda+k} + \frac{\mathbb{I}[k \leq n-\lambda]}{2h} \right] \\
& \quad \left. + \binom{h+k-2}{k} \frac{\mathbb{I}[k \leq n-\lambda]}{2h} \right\} \\
= & \sum_{h=1}^{\lfloor \lambda/2 \rfloor} \sum_{k=0}^{\lambda-2h} \alpha^{\lambda-h-k}(1-\alpha)^{h+k-1} \cdot \binom{\lambda-h-k-1}{h-1} \cdot 2 \cdot \\
& \left\{ \binom{h+k-1}{k-1} \frac{\mathbb{I}[k > n-\lambda]}{n-\lambda+k} + \binom{h+k-1}{k} \frac{\mathbb{I}[k \leq n-\lambda]}{2h} \right\} \\
= & \sum_{h=1}^{\lfloor \lambda/2 \rfloor} \sum_{k=0}^{\lambda-2h} \alpha^{\lambda-h-k}(1-\alpha)^{h+k-1} \binom{\lambda-h-k-1}{h-1} \frac{\psi(k)}{h+k} \binom{h+k}{h} \\
= & \sum_{t=1}^{\lambda-1} \frac{\alpha^{\lambda-t}(1-\alpha)^{t-1}}{t} \sum_{h=1}^{\min\{t, \lambda-t\}} \binom{t}{t-h} \binom{\lambda-t-1}{h-1} \psi(t-h).
\end{aligned} \tag{A.12}
$$

(A.12) is the polynomial-time algorithm of computing $\Pi(\mathbb{D}_{\mathbf{I}})$. The tradeoff relationships of different settings of (n, λ) is shown in Figure 7. Because of $\psi(t-h)$, the summation above is difficult to simplify. Thus, we apply $\psi(t-h) \leq 2$ and provide a lower bound for privacy metric.

$$
\begin{aligned}
& 1 - \Pi(\mathbb{D}) - \frac{2(1-\alpha)^L}{n} \\
\leq & \, 2 \sum_{t=1}^{\lambda-1} \frac{\alpha^{\lambda-t}(1-\alpha)^{t-1}}{t} \sum_{h=1}^{\min\{t, \lambda-t\}} \binom{t}{t-h} \binom{\lambda-t-1}{h-1} \\
= & \, 2 \sum_{t=1}^{\lambda-1} \frac{\alpha^{\lambda-t}(1-\alpha)^{t-1}}{t} \binom{\lambda-1}{t-1} \\
= & \, \frac{2}{\lambda} \sum_{t=1}^{\lambda-1} \alpha^{\lambda-t}(1-\alpha)^{t-1} \binom{\lambda}{t} \\
= & \, \frac{2}{\lambda(1-\alpha)} \left[1 - \alpha^\lambda - (1-\alpha)^\lambda \right].
\end{aligned}
$$

The lower bound in (4.5) is justified. □

A.6 Proof of Theorem 5.1

Recall that the PCN is run as a Markov chain. In 3 steps, we prove the success rate tends to 0 when the Markov chain becomes stable after infinitely many transactions.

(1) There exists an absorbing state.
For an edge (u, v) with $0 \leq b(u, v), \tilde{b}(v, u) < \ell$, it's impossible to route through this edge via any direction. So we call it **locked**.
In $\mathcal{G}$, we pick a set of edges $\mathcal{E}' \subseteq \mathcal{E}$ and set them locked, such that in the (undirected) graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E} - \mathcal{E}')$, for any (u, v) that is possible to be chosen as the pair of sender and recipient, u, v belong to different connected components. Note that such $\mathcal{E}'$ exists: $\mathcal{E}'$ could be $\mathcal{E}$, where all nodes are separated from each other.

(2) There's a non-zero probability that any transient state is transitioned to an absorbing state within finite steps.
In an transient state $\mathcal{G}_0$ (excluding locked edges), there exists a pair of nodes (u, v) which are still connected via some path $\mathcal{P}$. As assumed, there exists an edge $\varepsilon \in \mathcal{P}$ whose public balance is not updated with probability at least $\beta > 0$. Assume it takes k_1 transactions of value ℓ to deplete ε on the direction to v. After this, it takes another k_2 transactions of value ℓ to deplete ε on the opposite direction. Assume probability of choosing (u, v, ℓ) and (v, u, ℓ) are p_1 and p_2, respectively. Then the probability of locking ε after $k_1 + k_2$ transitions is at least $(\alpha p_1)^{k_1}(\beta p_2)^{k_2}$, which is positive.
Let $\mathcal{G}_1$ denote the new state after locking ε. Note that $\mathcal{G}_1$ has one fewer edge than $\mathcal{G}_0$. Since the number of edges is finite, there's a positive probability that $\mathcal{G}_0$ is transitioned to a state where no pair of nodes is connected via a path of unlocked edges. In this case, no more edge can be locked, and this is exactly an absorbing state discussed above.

(3) Success rate reaches 0 almost surely as $T \to \infty$.
Apparently, in any absorbing state, no transaction sampled from the same distribution can succeed. In this Markov chain, at least one absorbing state is reachable from any transient state. Thus, almost surely, the PCN will be at an absorbing state after finite number of transactions. Because such PCNs will no longer be able to support any transaction sampled from the same distribution, the overall success rate is 0. □

B SIMULATION RESULTS

B.1 Transaction Value

In the experimental settings in Section 6, the value of every transaction followed a Pareto distribution $\Psi(1.16, 1000)$. To study the curves for different transaction value distributions, we first selected two typical network structures: Lightning Network and Erdős-Rényi network with the same size and density. We also fixed the size of the workload to 3,000 transactions. We considered two properties of transaction values: distribution type and expectation. The distribution types include the uniform distribution from 0 to twice the expected value, and 3 different Pareto distributions with $\beta = 1.1, 1.16, 1.25$. Their PDFs are plotted in Figure 16 (Appendix B). Figure 13 illustrates the privacy-success rate tradeoffs for each of these experimental settings. Here, the "imbalanced" curve in the top row refers to our approach for generating the endpoints of transactions; instead of choosing the source and destination uniformly, we assign users a weight of 1 with probability 0.8 and 16 with probability 0.2. When generating transactions, we sample endpoints with probability proportional to their weights. On average, 20% of the users hold 80% of the weight; this models the imbalance observed in real financial transactions. These plots show that when the mean transaction value increases, the success rate decreases. This is because we are keeping the capacity distribution fixed, so larger transactions are more likely to fail. Similarly, using increasingly heavy-tailed Pareto

transactions (smaller β) causes the minimum transaction value to decrease, which increases the success rate. Although these plots illustrate some trends related to transaction value distribution, the main takeaway is the fact that for a variety of transaction value distributions, the slope of the privacy-success rate curves is shallow. Figure 11 also shows the slopes of curves are also not decisively steep when the workloads are heavier. This implies that **one cannot trade small losses in privacy for large gains in success rate, or vice versa.**

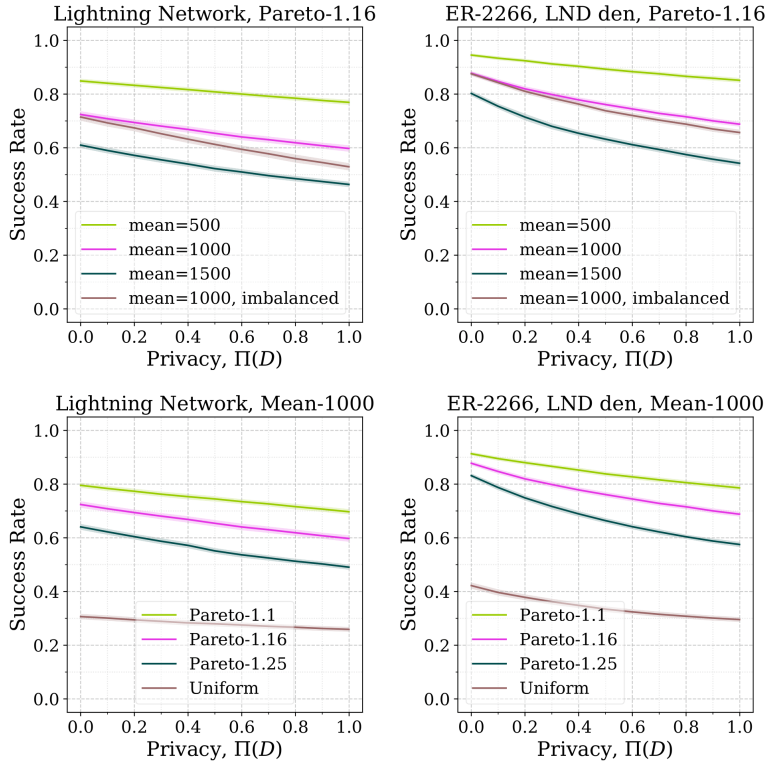

Fig. 13. Success rate-privacy tradeoffs for different transaction value distributions. The top row considers Pareto-distributed transactions of different means; the bottom row considers different distributions with the same mean.

B.2 Workload size

We next investigate the role of workload size by varying the number of transactions in our workload. Figure 14 shows the average success rate as a function of privacy as we process from 1,000 to 100,000 transactions. The ER graph has a density of $\log(50)/50$. More precisely, this experiment revealed two unexpected phenomena: (1) for all privacy levels (including zero), success rate decreases as we process more transactions. It is not obvious *a priori* why this should happen, as transactions are generated uniformly at random between nodes; on average, transaction flows should be balanced. (2) Success rate is not always monotonically decreasing in the privacy parameter. This phenomenon is triggered by "deadlocks" introduced in Section 5. These deadlocks suggest that sacrificing a little privacy can actually *hurt* average success rate in the long term. Hence system designers should be

wary of deploying a noise mechanism that sacrifices only a little privacy; not only are the gains modest in theory, they can actually be negative in practice, unless system designers deal with deadlocks.

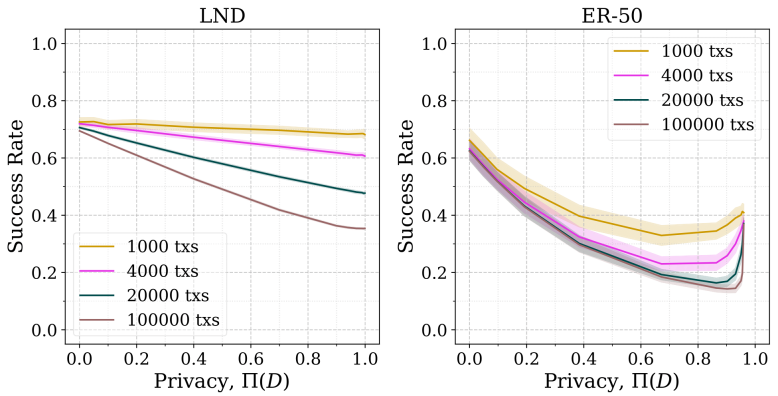

Fig. 14. Transaction workload size vs success rate.

Declining success rate over time. The declining success rates are partially due to the previously-described deadlocks. However, we observe the phenomenon even at zero privacy, when deadlocks cannot happen. The reason may be related to the formation of *routing bottlenecks*. Suppose a node v receives many transactions until all of v's neighboring channels are imbalanced, with most of the balance on v's side of the channel. In such a scenario, nobody can route through v: funds cannot reach v, so it cannot relay money. Under randomized transaction workloads like ours, such bottlenecks are theoretically recurrent; in practice, they can also occur if some nodes are popular money recipients (e.g., merchants). The only way to break a bottleneck is for the bottlenecked node(s) to send their own transactions out of the isolated zone. In a large network, the likelihood of this happening for a uniform workload is small. Hence, the system may be bottlenecked for a long time.

B.3 Network Size

To observe the effects of network size, the source and destination of each transaction are still drawn uniformly, and the values of transactions still follow Pareto distribution $\Psi(1.16, 1000)$. To generate smaller versions of the Lightning network, we snowball sample the full-sized graph until reaching the desired size. For each network size, we sample 100 networks, and the average density of both Erdős-Rényi and LND networks are matched.

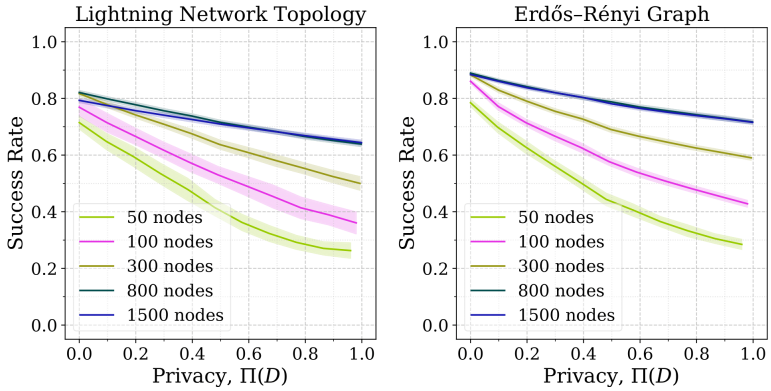

Fig. 15. Privacy-Success rate tradeoff curves for different graph sizes.

The tradeoff curves are shown in Figure 15. When the size is small, there is high variance in network topology of LND, which leads to high variance in success rates. As the network size increases, the privacy-success rate tradeoff becomes insensitive to these variations. Moreover, we observe that success rate increases. This may be because as more nodes are added, there are more paths for transactions to traverse, which gives additional robustness to link imbalance.

B.4 Data distributions and workloads

Figure 16 illustrates the pdfs of the Pareto distributions used in our experiments. Notice that Pareto distributions have a minimum transaction value, and are also heavy-tailed.

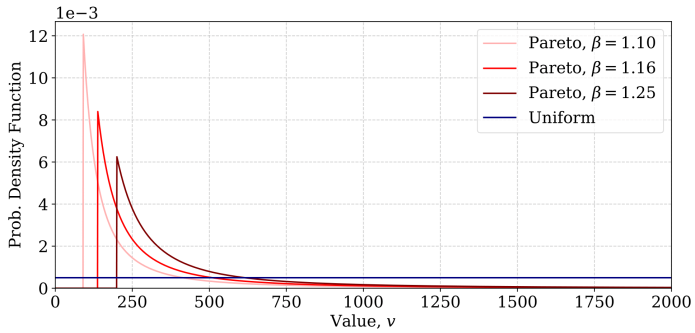

Fig. 16. Pareto distribution pdfs for different parameter settings.

Our experiments use a snapshot of the real Lightning network topology, illustrated in Figure 17. The graph has many nodes of degree 1 or 2, with several large hubs that are well-connected. This is qualitatively different from the synthetic ER graphs we also used in experiments.

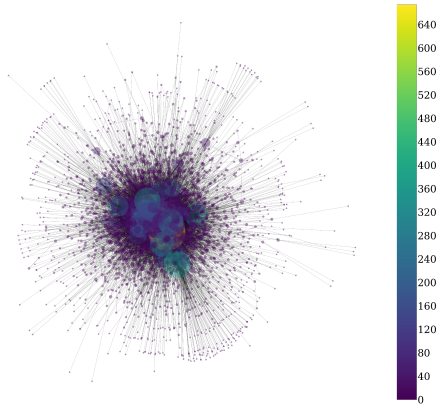

Fig. 17. Snapshot of the Lightning network topology from December 28, 2018. Node sizes are scaled proportionally to their degree in the graph.

Received January 2020; revised February 2020; accepted March 2020