DP-Mix: Differentially Private Routing in Mix Networks

Mahdi Rahimi
COSIC, KU Leuven
Leuven, Belgium
mahdi.rahimi@esat.kuleuven.be

***Abstract*—Mixnets, as overlay networks, ensure anonymity for messages by forwarding them through intermediary nodes that obscure their traffic patterns from network-level adversaries. Nonetheless, the selection of intermediaries is traditionally performed uniformly at random, resulting in optimal routes being chosen no more frequently than suboptimal ones. This often causes messages to traverse inefficient paths that degrade performance or weaken security. While there have been proposals to improve route selection in mixnets, they are limited to latency reduction and rely on heuristic strategies that lack formal anonymity guarantees.**

To bridge this gap, we develop a framework for differentially private routing in mixnets aimed at general-purpose optimization. In this framework, each candidate route is assigned an optimality score, and routes are then selected to favor high-scoring paths while preserving anonymity under a pure-ε differential privacy guarantee. We instantiate this model for optimizing path security, enhancing reliability, and minimizing communication latency. Additionally, we introduce a gradient-based algorithm applied post hoc to the route selection process—without weakening privacy guarantees—to prevent the over-selection of particular nodes, which could otherwise lead to security vulnerabilities or network congestion. Through analytical evaluation and simulation over data from deployed Nym mixnets, we demonstrate that DP-Mix consistently achieves high optimality scores across all instantiations while preserving a strong level of anonymity. In particular, for latency optimization, our method outperforms state-of-the-art solutions, achieving up to a $8\times$ improvement in the latency–anonymity trade-off.

***Index Terms*—Mixnets, Anonymity, Strategic Routing, Differential Privacy.**

1. Introduction

Mix networks (*mixnets*) are anonymity networks that provide strong privacy guarantees [1]–[4] for clients against a Global Passive Adversary (GPA)—capable of observing all communication exchanges within the network. Specifically, mixnets operate as overlay networks designed to enhance anonymity by routing client messages through multiple intermediary hops, known as *mixnodes* [5], before delivering them to their final destinations. Additionally, messages received at any mixnode are shuffled together, ensuring that input messages to the mixnode cannot be directly correlated with their output counterparts [4]. Under this setting, as long as at least one mixnode along a message's path behaves honestly, linking input and output traffic remains computationally infeasible [6], effectively thwarting the GPA.

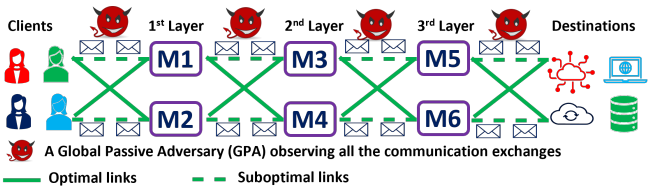


Figure 1: Differentially private routing for a stratified mixnet with $L = 3$ mixing layers, each hosting $W = 2$ mixnodes. Solid lines represent optimal links, while dashed lines indicate suboptimal links.

Mixnets, on the other hand, can be designed in various ways depending on two primary factors: (1) the *shuffling process*, which determines how messages are delayed or mixed within each mixnode, and (2) the *mixnet topology*, which defines how mixnodes are arranged in the network. Among the various designs, we focus on a *stratified* topology with the mixing strategy set to *stop-and-go mixing* [7]. This design, widely known as *Loopix* [3], is illustrated in Fig. 1, where the mixnet is partitioned into L distinct layers, and message paths are constructed by selecting one mixnode from each layer. Additionally, each mixnode, following the stop-and-go mixing principle, flushes received traffic after a delay sampled from an exponential distribution. Limiting our study to this structure, however, is due to its practical advantages, as demonstrated by its adoption in deployed mixnets such as Nym [4].[1] That said, our approach can be readily adapted—with moderate changes—to other mixnet architectures as well.[2]

Problem Identification. In mixnets, message routes are typically constructed by selecting one node from each layer uniformly at random, leading to fully randomized path selection. While this approach is simple and widely used, it poses the risk of routing messages through suboptimal paths, since all paths—optimal or not—are sampled with equal

1. https://nym.com
2. See Appendix F for more details.

probability. For example, a suboptimal path may involve mixnodes that are unreliable (e.g., going offline intermittently), geographically distant (resulting in high end-to-end communication latency), or concentrated in a single jurisdiction—thereby compromising anonymity in the presence of localized adversaries. In such cases, prioritizing the selection of optimal paths that mitigate these risks proves advantageous, leading to improved overall routing performance.

Adversarial Threats. Improved routing performance must not compromise the considerable anonymity provided by the mixnet, nor introduce new vulnerabilities that adversaries could exploit. Specifically, we consider two adversarial models against which the routing strategy must remain resilient: (1) A *Global Passive Adversary (GPA)* capable of observing all communication links in the network, but unable to observe the internal mixing processes performed at each mixnode. Such an adversary may attempt to infer a probabilistic mapping between incoming and outgoing messages across the network. (2) A *mixnode adversary*, which can compromise a subset of mixnodes and trivially link messages entering those nodes to their outputs, aiming to deanonymize full message routes.

Design Goals. Given the limitations of current routing strategies in mixnets, our primary goal in DP-Mix is to develop a general-purpose strategic routing framework that enables clients to select message paths based on specific optimization objectives—such as avoiding unreliable links, minimizing latency, or improving path security—while ensuring that anonymity is preserved under the aforementioned adversarial models to a desirable extent, and that any potential anonymity loss is quantifiable.

Contributions. Under the specified design goals, DP-Mix provides the following contributions. **First**, we introduce the concept of *Differentially Private Routing (DPR)*, where each path in the mixnet is assigned an optimality score, and path selection is performed probabilistically using the exponential mechanism [8], which supports pure ε-differential privacy [9]. This mechanism probabilistically favors higher-scoring (i.e., more optimal) paths, while still allowing less optimal ones to be selected with lower probability. The parameter $\varepsilon \geq 0$, which governs this mechanism, controls the privacy–optimality trade-off: smaller values provide stronger privacy (i.e., more randomness), while larger values prioritize paths with higher optimality scores. Formally, in the context of differential privacy, our mechanism guarantees that for any two mixnet scenarios where all client–destination pairs are identical except for one client's destination, the probability of assigning any path set to client–destination connections in one scenario is at most a factor of e^{ε} larger than in the other. This ensures that it remains difficult for the GPA to distinguish between the two scenarios, thereby preserving the anonymity of the client whose destination differs in these scenarios.

Second, we provide three practical instantiations of DPR for concrete routing objectives: (1) *Latency optimization*, with the goal of minimizing end-to-end message latency. (2) *Unreliability avoidance*, which steers path selection away from routes composed of mixnodes that are prone to going offline or becoming unstable. (3) *Jurisdiction-aware routing*, which promotes diversity of paths across legal jurisdictions to mitigate the risk of localized adversarial surveillance.

Third, we introduce the *Traffic Alignment Mechanism (TAM)* to control the selection bias introduced by DPR. Notably, without such regulation, DPR may cause some nodes to receive disproportionately high traffic, which can lead to two issues: (1) Congestion, latency spikes, or message drops in resource-constrained mixnets. (2) Increased vulnerability in adversarial settings, where a mixnode adversary may attempt to introduce nodes with higher processing capacity to deanonymize a larger fraction of messages. To address these challenges, TAM employs a gradient-based adjustment procedure that modifies the selection probabilities of succeeding nodes to minimize imbalance. Crucially, TAM is applied as a post-processing step and does not affect the formal privacy guarantees provided by the DPR.

Fourth, we implement a complete prototype of DP-Mix in *Python* (approximately 5000 LOC), supporting both analytical and simulation-based evaluation of the DPR framework and TAM. The prototype also includes two state-of-the-art routing approaches—LARMix [10] and LAMP [11]—which are designed exclusively for latency optimization using heuristic methods and are unable to satisfy the pure ε-differential privacy definitions (see § 7 for details on related work). Our evaluation is primarily conducted using data collected from the deployed Nym mixnet [4]. However, to simulate mixnet scenarios with broader geographic diversity, we also incorporate the RIPE Atlas dataset [12], which provides a larger number of globally distributed measurement nodes.[3]

Furthermore, our results show that DPR consistently achieves a favorable balance between routing optimality and anonymity. For example, at the cost of a small reduction in anonymity, measured as Shannon entropy [13] (e.g., 0.1 or 0.2 bits), DPR increases the average reliability or jurisdictional diversity by 40% and 70%, respectively, for the unreliability-avoidance and jurisdiction-aware routing instantiations. In the case of latency optimization, we observe a similar trend. Notably, DPR significantly outperforms LARMix [10] and LAMP [11], achieving up to 94% latency reduction while preserving up to 1 bit higher anonymity. **Finally**, we evaluate the resilience of DPR against mixnode adversaries under two mixnode corruption strategies: (1) *Random corruption*, and (2) *Greedy corruption*. Our results show that even under these attack models, the fraction of fully compromised paths remains negligible across all DPR instantiations, demonstrating the robustness of our approach under such adversarial conditions.

2. Approach

In this section, we detail the methodology of DP-Mix, beginning with the DPR mechanism and its instantiations for different optimization goals, and concluding with the TAM

3. You can access the full implementation, along with guidelines to reproduce the results, at https://github.com/DPMix/DP-Mix.

algorithm, which mitigates load imbalances introduced by the DPR-based routing strategies.

2.1. Differentially Private Routing (DPR)

To construct a message route in mixnets, clients traditionally select one of the available paths uniformly at random. For instance, in Fig. 1, each client has 8 possible paths to choose from and selects each with equal probability $\frac{1}{8}$. This design choice was originally motivated by its simplicity and its theoretical guarantee of maximizing confusion for a GPA observing network traffic in an attempt to infer message routes. However, uniform path selection fails to discriminate between optimal and suboptimal routes. For example, a suboptimal path may incur higher communication latency, consist of unreliable mixnodes, or be composed of nodes located within a single legal jurisdiction—thus weakening either the performance or security provided by the mixnet. To address this, we introduce the DPR strategy, which improves network performance or security by probabilistically favoring highly optimal paths, while still providing formal guarantees on anonymity loss, since it is governed by the exponential mechanism [8], supporting pure ε-differential privacy guarantees [9].

To understand DPR, consider the scenario illustrated in Fig. 1, where clients connect to a stratified mixnet consisting of L layers, each containing W mixnodes. In such a scenario, the total number of possible end-to-end paths for each client $\mathsf{c} \in \mathcal{S}_C$ (the set of clients), communicating with a destination $\mathsf{d} \in \mathcal{S}_D$ (the set of destinations), is W^L. We define the set $\mathcal{S}_P^{\mathsf{c}}$ as the set of all valid paths from client c to destination d. Each path $P_i^{\mathsf{c}} \in \mathcal{S}_P^{\mathsf{c}}$ is associated with a utility value ν_i^{c}, where a higher value indicates better performance or security.

Naturally, each client prefers to select the most optimal path, denoted as $P_{i^*}^{\mathsf{c}}$, which maximizes their utility, i.e., $\nu_{i^*}^{\mathsf{c}}$. However, deterministically choosing the highest-utility path allows a GPA to infer the complete message route and compromise the client's anonymity by linking it to its destination. To mitigate this risk, leveraging the exponential mechanism [8], we introduce randomness into the path selection process, enabling clients to probabilistically favor higher-utility paths while maintaining a quantifiable level of anonymity.

Formal Representation. To provide a more formal representation of DPR in the context of differential privacy, consider the dataset $X \in \mathcal{X}^n$ as a collection of tuples, where each tuple represents a client and its intended destination, and let the total of n client–destination pairs communicate through the mixnet. This dataset X is highly sensitive, as it contains the exact mapping of clients to their communication targets. Revealing such information without anonymization would violate user privacy—highlighting the core motivation for using mixnets in the first place. However, if clients in mixnets employ a naive routing strategy that deterministically selects the highest-utility path $P_{i^*}^{\mathsf{c}}$, their connection to the destination may become linkable and thus susceptible to inference by a GPA.

To address this risk, we apply the *exponential mechanism* [8] for route selection—a fundamental tool in the differential privacy literature [9]—which probabilistically maps a sensitive dataset X, representing client–destination pairs, to an output $h \in \mathcal{H}$, where $\mathcal{H}$ denotes the space of possible outcomes (e.g., path selections). Additionally, this mechanism relies on a scoring function $s : \mathcal{X}^n \times \mathcal{H} \to \mathbb{R}$, where higher scores correspond to more desirable outcomes. The output of the mechanism is a probability distribution over all elements in $\mathcal{H}$, biased toward high-scoring configurations, while ensuring compliance with ε-differential privacy. A formal definition and the associated privacy guarantees are provided in Appendix B. In the following, we adapt this mechanism to develop the DPR framework.

To develop DPR based on the exponential mechanism, we denote the set of objects as $\mathcal{H} = \{h_j \mid 1 \leq j \leq (W^L)^{|\mathcal{S}_C|}\}$, where each object h_j is a configuration that assigns one path to each client. That is, h_j maps each client-destination pair—i.e., $\mathsf{c} \in \mathcal{S}_C$, $\mathsf{d} \in \mathcal{S}_D$—to a path $P_i^{\mathsf{c}} \in \mathcal{S}_P^{\mathsf{c}}$. Since each client has a path set of size $|\mathcal{S}_P^{\mathsf{c}}| = W^L$, and there are $|\mathcal{S}_C|$ clients, the total number of configurations is $|\mathcal{H}| = (W^L)^{|\mathcal{S}_C|}$. Among these configurations, there exists at least one optimal object $h^* \in \mathcal{H}$, which assigns the highest-utility path to every client:[4]

$$h^* = \left\{ P_{i^*}^{\mathsf{c}} \in \mathcal{S}_P^{\mathsf{c}} \mid \nu_{i^*}^{\mathsf{c}} = \max_i \nu_i^{\mathsf{c}},\ \forall\, \mathsf{c} \in \mathcal{S}_C,\ \mathsf{d} \in \mathcal{S}_D \right\}.$$

Moreover, we define the scoring function $s(X, h_j)$ for the exponential mechanism as described in Eq. (1), representing the weighted average utility across all clients. This score is computed by aggregating the utility values ν_i^{c} assigned to the route selected by each client $\mathsf{c} \in \mathcal{S}_C$, under the path configuration specified by h_j. In Eq. (1), the weight a_{c} corresponds to the volume of traffic generated by client c.[5]

$$s(X, h_j) = \sum_{\mathsf{c} \in \mathcal{S}_C} a_{\mathsf{c}} \cdot \nu_i^{\mathsf{c}}. \tag{1}$$

Using this score function, the exponential mechanism selects a configuration $h_j \in \mathcal{H}$ with probability proportional to: $\Pr[M(X) = h_j] \propto \exp\left(\frac{\varepsilon \cdot s(X, h_j)}{2\Delta}\right)$, where Δ is the sensitivity of the scoring function s—i.e., the maximum change in score that can result from modifying a single entry in the dataset (i.e., one client-destination pair), across all configurations in $\mathcal{H}$. This formulation introduces controlled randomness, allowing higher-utility configurations to be selected more frequently while preserving differential privacy.

The privacy parameter ε, on the other hand, governs the trade-off between utility and anonymity. When $\varepsilon = 0$, all configurations are equally likely to be selected, thereby maximizing anonymity but minimizing performance. As ε

4. Multiple such objects may exist if clients have more than one equally optimal path.

5. In the case of uniform traffic generation across all clients, we set $a_{\mathsf{c}} = \frac{1}{|\mathcal{S}_C|}$. The uniform traffic generation assumption arises because clients typically generate dummy traffic alongside real traffic for anonymity purposes, making the amount of traffic generated by each client appear similar.

increases, the mechanism increasingly favors higher-utility configurations, improving performance at the expense of reduced anonymity. Nonetheless, the exponential mechanism guarantees formal privacy under pure ε-differential privacy.

This guarantee formally states that if there exist two datasets $X, X' \in \mathcal{X}^n$ differing in exactly one entry—commonly referred to as neighboring datasets (e.g., differing by a single client-destination pair)—then the exponential mechanism M satisfies: $\frac{\Pr[M(X)=h]}{\Pr[M(X')=h]} \leq \exp(\varepsilon), \quad \forall h \in \mathcal{H}$, ensuring that the presence or absence of any individual client-destination pair has only a limited influence on the output distribution. This property provides a formal anonymity guarantee.[6]

Differential Privacy Implications in Mixnets. To better understand the pure ε-privacy guarantee in mixnets, consider two scenarios where all client–destination pairs are fixed except for client c_1, who connects to destination d_1 in the first scenario and d_2 in the second. Let the first scenario be represented by X and the second by X'. The DP guarantee states that the probability of assigning specific path sets in Scenario 1 compared to Scenario 2 is at most e^{ε} larger. This limits the ability of the GPA to distinguish whether c_1 is communicating with d_1 or d_2, particularly when ε is small.

That said, the privacy budget ε decreases as the number of exchanged messages between clients and their destinations increases in the mentioned scenarios. Specifically, if sending a single message guarantees ε-DP, then sending K messages reduces the overall privacy guarantee to $K\varepsilon$ (pure-DP) under basic composition. However, under advanced composition [14], the privacy loss can be bounded as $\varepsilon \approx \sqrt{K}\varepsilon$, though in this case the guarantee follows (δ, ε)-DP rather than pure-DP.

Lastly, note that although our presentation of DP-Mix is under the global differential privacy setting, it can also be applied under local differential privacy, where each client uses the exponential mechanism to select a path. In this case, the score function might need to be slightly adapted to capture the performance of the selected path for each client individually rather than for the overall network. Local differential privacy may also yield lower performance compared to global differential privacy, but this can be mitigated by increasing ε.

Scalability of DPR. We emphasize that the path space in the exponential mechanism for a mixnet with L layers, each containing W mixnodes, is W^L. Thus, the overall complexity of such an approach is $O(W^L) \approx O(N^3)$, as typically $L = 3$ and $N = LW$. We note that W is on the order of a few hundred, which makes this overhead reasonable. Under resource-constrained scenarios, however, a client may construct the path by selecting one hop at a time, instead of the full path at once, by applying DPR iteratively. This corresponds to running DPR L times, each with a path space of W, thereby reducing the complexity to $O(LW) = O(N)$, albeit at the cost of reduced performance optimality.

6. For further details on the exponential mechanism and pure ε-differential privacy, see Appendix B.

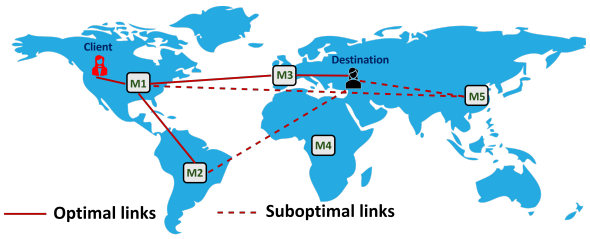


Figure 2: Latency optimization: A naive route selection leads to a suboptimal path like $M_1 \rightarrow M_5$ for a given destination, whereas DPR enables the selection of a highly optimal path such as $M_1 \rightarrow M_3$.

State Information for Applying DPR. Clients require network configuration and related information to assign scores to paths when using DPR methods. Since DP-Mix builds on the Nym mixnet, the network information and the process of updating path scores are grounded in protocols already employed by Nym. In particular, the Nym network reconfigures its topology (the arrangement of nodes into layers) every few hours. Before each reconfiguration, Nym measures internode latencies, geolocations, and reliability using the VerLoc protocol [15] in a fault-tolerant and decentralized manner. Consequently, prior to each reconfiguration, Nym provides clients with updated network state information. Based on this information and the optimization objectives, path scores are then updated and securely selected.

Flexibility of DPR. Finally, we note that DPR has the potential to be applied across a wide range of settings. On one hand, the scoring function and associated objectives are defined in a highly general form, enabling the adoption of diverse utility functions. On the other hand, although the object space is defined over complete client-to-destination paths, it can be flexibly reconfigured as needed. For instance, if optimization is desired only over the segment from the client to the final mixnode (rather than to the destination), DPR can naturally accommodate such cases by evaluating the utility of the path prefix terminating at the last hop. Moreover, in scenarios where the destination set $\mathcal{S}_D$ is fixed, the destination can be incorporated into the optimization process. This practical flexibility addresses a key limitation of prior work such as LAMP [11], which restricts communication latency optimization to the segment between the client and the last hop. We elaborate further on this point below while illustrating the latency optimization instantiation of DPR. Beyond these interesting features, the DP-Mix approach can potentially be applied to other anonymous communication systems, such as Tor [16], I2P[7], and network-level anonymity mechanisms in blockchain protocols [17], [18]. See Appendix E for a broader discussion. That said, in this paper, we focus on the instantiation of DP-Mix within the context of mixnets, as detailed below.

Latency Optimization. In mixnets, messages are routed

7. https://geti2p.net

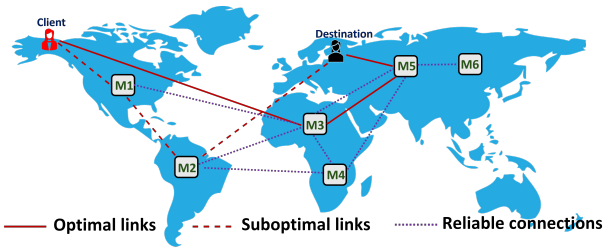


Figure 3: Unreliability-Avoidance: Dotted lines represent reliable connections between mixnodes. A naive routing approach may select a path such as $M_1 \rightarrow M_2$, which suffers from poor connectivity. In contrast, a DPR-based strategy selects a reliable route, such as $M_3 \rightarrow M_5$.

through multiple hops, which may be geographically distant, thereby introducing high link latency—an undesirable property for latency-sensitive applications such as web browsing. To mitigate this, we instantiate DPR with the objective of optimizing end-to-end latency. Given $X \in \mathcal{X}^n$ and the set $\mathcal{S}_P^{\mathrm{c}}$ for each $\mathrm{c} \in \mathcal{S}_C$, we focus on two concrete instantiations of this setting: (1) *Client-aware routing*: $\mathcal{S}_P^{\mathrm{c}}$ is defined as the set of all paths from client c to the last hop in the mixnet, similar to the setup used in LAMP [11]. (2) *Client-destination-aware routing*: When the destination set $\mathcal{S}_D$ is given, $\mathcal{S}_P^{\mathrm{c}}$ is defined as the set of full end-to-end paths from each client to their respective destination.

In both mentioned cases, we define the latency-based utility for a path as follows. For a client c and path $P_i^{\mathrm{c}} \in \mathcal{S}_P^{\mathrm{c}}$, the utility is given by: $\nu_i^{\mathrm{c}} = -l_i^{\mathrm{c}}$, where l_i^{c} denotes the total link latency incurred along path P_i^{c}, either up to the final mixnode or along the complete path to the destination, depending on the optimization scope. The global score function is then computed using Eq. (1), where the sensitivity Δ corresponds to the maximum change in latency over all configurations, weighted by the client traffic distribution. Assuming uniform traffic generation, we set $\Delta = \frac{1}{|\mathcal{S}_C|}$.[8] Finally, the exponential mechanism selects each configuration $h_j \in \mathcal{H}$ with probability proportional to: $\exp\left(\frac{\varepsilon|\mathcal{S}_C|\cdot s(X,h_j)}{2}\right)$, thereby favoring lower-latency configurations while preserving formal guarantees under pure ε-differential privacy.

Moreover, to illustrate this instantiation more concretely, Fig. 2 presents an example in which a client aims to establish a message route for a given destination. A naive strategy may result in selecting a suboptimal path such as $M_1 \rightarrow M_5$, leading to high latency. In contrast, with DPR-based latency optimization, the client may instead select a path like $M_1 \rightarrow M_2$ (optimized up to the last mixnode) or $M_1 \rightarrow M_3$ (optimized end-to-end), both of which yield significantly improved latency performance.

Unreliability-Avoidance. Beyond latency optimization, DPR can also be applied to avoid unreliable connections in mixnets. Specifically, mixnets consist of globally distributed mixnodes that often vary significantly in reliability—i.e., their ability to remain responsive and connected during message transmission. Unreliability may arise from intermittent node outages due to infrastructure limitations, adverse regional network conditions, or deliberate censorship attempts. When combined with uniform random path selection, this results in a non-negligible probability that selected paths include unreliable nodes, leading to message loss or delivery delays. To address this issue, we instantiate DPR to favor paths composed of highly reliable mixnodes. To the best of our knowledge, this is the first approach that enables reliability-aware path selection in mixnets, and it can be directly integrated with deployed systems such as Nym to improve robustness and overall performance.

Formally, for the defined dataset $X \in \mathcal{X}^n$ and the set of candidate paths $\mathcal{S}_P^{\mathrm{c}}$ for each client c, each path $P_i^{\mathrm{c}} \in \mathcal{S}_P^{\mathrm{c}}$ is evaluated using a reliability-based utility value: $\nu_i^{\mathrm{c}} = \min\left(\phi_{1i}^{\mathrm{c}}, \phi_{2i}^{\mathrm{c}}, \ldots, \phi_{Li}^{\mathrm{c}}\right)$, where ϕ_{ji}^{c} denotes the reliability of the j-th mixnode in the path P_i^{c}, typically measured as the fraction of successful connections that the node maintains relative to the maximum number of possible connections. This utility reflects the worst-case reliability along the path and encourages selection of paths that avoid unreliable mixnodes.

In this case, the overall score function is computed using Eq. (1), where the sensitivity Δ is determined by the maximum change in utility resulting from modifying a single client-destination tuple. Assuming uniform traffic generation (i.e., $a_{\mathrm{c}} = \frac{1}{|\mathcal{S}_C|}$) and noting that reliability values are bounded within $[0, 1]$, we have $\Delta = \frac{1}{|\mathcal{S}_C|}$. Finally, using the exponential mechanism, each configuration $h_j \in \mathcal{H}$ is selected with probability proportional to: $\exp\left(\frac{\varepsilon|\mathcal{S}_C|\cdot s(X,h_j)}{2}\right)$, ensuring that more reliable paths are probabilistically favored while preserving the formal guarantees of pure ε-differential privacy.

Furthermore, as an example, Fig. 3 illustrates a scenario in which connectivity between mixnodes is represented by dotted lines. Under a naive approach, a client may select a path such as $M_1 \rightarrow M_2$ to reach the destination; however, due to the poor reliability of these nodes, the resulting route is unstable and prone to failure. In contrast, with a DPR-based strategy, the client can select a more reliable path such as $M_3 \rightarrow M_5$, thereby improving delivery robustness.

Jurisdiction-Aware Routing. Lastly, we consider scenarios in which the security of mixnets is compromised due to uniform path selection resulting in message routes composed entirely of nodes within a single legal jurisdiction. This typically occurs when the mixnet topology is dominated by nodes hosted in a single regulatory domain. In such cases, uniformly random path selection may yield routes that lack jurisdictional diversity, thereby reducing anonymity and increasing vulnerability to localized adversaries. To mitigate this, we instantiate DPR to probabilistically favor paths that traverse a greater number of distinct jurisdictions. Specifically, for each path P_i^{c} available to client $\mathrm{c} \in \mathcal{S}_C$, we define the utility score as: $\nu_i^{\mathrm{c}} = \mathcal{J}_i^{\mathrm{c}}$, where $\mathcal{J}_i^{\mathrm{c}}$ denotes the number of distinct jurisdictions represented by the mixnodes

8. Latency values are measured in milliseconds and typically remain below one second in our evaluation.

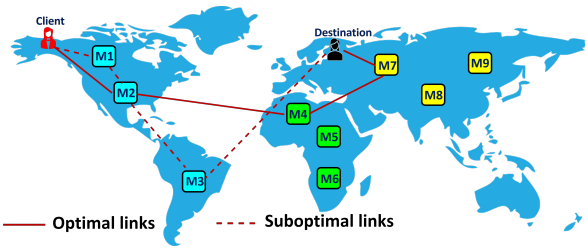


Figure 4: Jurisdiction-Aware Routing: blue, green, and yellow nodes represent distinct jurisdictions. The suboptimal path (e.g., $M_1 \to M_2 \to M_3$) includes only nodes from a single jurisdiction, while the optimal path (e.g., $M_2 \to M_4 \to M_7$) spans multiple jurisdictions, thereby enhancing anonymity.

along path P_i^c. In a stratified mixnet with L layers, we have $1 \leq \mathcal{J}_i^c \leq L$. A higher value of $\mathcal{J}_i^c$ implies greater jurisdictional diversity, and crossing at least two distinct jurisdictions is typically required to ensure strong unlinkability and resistance to jurisdiction-specific surveillance.

In this setting, the sensitivity of the scoring function can be derived as $\Delta = \frac{L-1}{|\mathcal{S}_C|}$, since the maximum possible change in jurisdictional diversity resulting from modifying a single client-destination pair is $L-1$, and this change is weighted by the number of clients $|\mathcal{S}_C|$, assuming uniform traffic generation across all clients. Using this definition in Eq. (1), the exponential mechanism selects configurations that probabilistically favor jurisdictionally diverse paths while preserving the formal guarantees of pure ε-differential privacy.

Moreover, Fig. 4 illustrates a representative example in which nodes from different jurisdictions are shown in blue, green, and yellow. In this scenario, a naive approach may lead to the selection of a suboptimal path such as $M_1 \to M_2 \to M_3$, consisting entirely of mixnodes located within a single jurisdiction—thereby increasing vulnerability to localized adversaries. In contrast, DPR enables the selection of an optimal path like $M_2 \to M_4 \to M_7$, which traverses multiple jurisdictions and thus significantly improves security.[9]

3. Traffic Alignment Mechanism (TAM)

DPR-based routing improves performance and security by probabilistically favoring more optimal paths. However, this beneficial bias can inadvertently cause certain mixnodes to be selected disproportionately more often than others. Such traffic imbalance introduces two critical challenges. First, in resource-constrained environments, over-selected mixnodes may become congested, exceeding their processing capacity and leading to increased delays or message loss. Second, in adversarial settings, an adversary may exploit this skew by introducing high-capacity mixnodes into the network, heightening the risk of deanonymizing client-destination pairs by routing most of the traffic through them.

9. See Appendix D for additional optimization scenarios with different objectives instantiated using DPR.

To mitigate these risks, we propose a general-purpose, gradient-based adjustment algorithm, termed *TAM*, which refines the output distribution produced by the DPR mechanism to promote more balanced traffic allocation across mixnodes. Specifically, given an initial set of path selection probabilities from DPR, TAM iteratively adjusts them to align the expected traffic load received by each node with a predefined target selection probability, which abstracts an ideal distribution of traffic across mixnodes, ensuring that each node, in aggregate, receives traffic proportional to its capacity.

To describe TAM in greater detail, we define p_{ij}^m as the probability of selecting node j in layer $m+1$, given that node i in layer m has already been selected. For generality, we assume that layer 0 corresponds to the client set, while layers 1 through L represent the mixnode layers in the mixnet. Under this formulation, the conditional transition probability is defined as: $p_{ij}^m = \mathbb{P}[n_j^{m+1} \mid n_i^m]$, where n_i^m denotes node i in layer m, which may represent a client or a mixnode.

Moreover, the value of p_{ij}^m is computed by marginalizing over the path distribution induced by the DPR output, summing over all paths in which the pair (n_i^m, n_j^{m+1}) appears in adjacent layers m and $m+1$. These values also are assembled into a matrix $P^m = [p_{ij}^m]$, which captures the transition probabilities from layer m to layer $m+1$. Under this setup, the fraction of total traffic received by node j in layer $m+1$ is given by: $\sum_i p_{ij}^m$. Alternatively, the distribution of the traffic fraction received by nodes in layer $m+1$ can be expressed as the vector $(P^m)^\top \mathbb{1}$, where $\mathbb{1}$ is a vector of ones with dimensionality equal to W.

Let q^{m+1} denote a target traffic distribution over layer $m+1$, where q_j^{m+1} specifies the desired fraction of total traffic assigned to node j, typically reflecting its processing capacity (for example, if the processing capacity of each node in the mixnet is the same, this distribution is uniform). TAM aims to minimize the discrepancy between the actual traffic distribution $(P^m)^\top \mathbb{1}$ and the target q^{m+1}. This discrepancy is measured using the squared Euclidean distance between the two distributions, denoted as the Cost function in Eq. (2).

$$\text{Cost}(m) = \mathsf{tr}\left[\left((P^m)^\top \mathbb{1} - q^{m+1}\right)\left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right]. \tag{2}$$

To minimize this cost function, the TAM algorithm applies a gradient descent procedure to iteratively update the entries of P^m, following the update rule provided in Lemma 1; see its proof in Appendix C.

Lemma 1. To minimize the cost function in Eq. (2), the transition matrix P^m can be updated via gradient descent. Specifically, the update rule for entry p_{ij}^m is:

$$p_{ij}^m \leftarrow p_{ij}^m - \lambda \cdot e^{\left[\mathsf{tr}\left[(S_{ij}^m)^\top \mathbb{1} \cdot \left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right]\right]},$$

where $\lambda > 0$ is the learning rate, and S_{ij}^m is a sparse matrix with a single nonzero entry at position (i, j), equal to p_{ij}^m.

The TAM algorithm, on the other hand, requires iterative updates to reduce the loss function until a convergence

threshold is met. For example, the stopping criterion can be defined such that the Cost function (Eq. (2)) is less than 0.1. This procedure should be applied across all layers of the mixnet. For a more detailed description, please refer to the full specification of TAM in Appendix A. Lastly, we note that TAM does not affect the anonymity loss guarantees provided by DPR. This holds because TAM operates solely as a post-processing adjustment over the output of DPR—meaning that the privacy of a dataset cannot be compromised by any transformation applied after the mechanism has been executed; see the proof of this statement in Appendix C.

4. Evaluation

In this section, we evaluate our instantiations of the DPR mechanism. To this end, we begin by describing the experimental setup and the evaluation metrics used, and eventually introduce the evaluation results which are mainly assessed against the GPA.

Experimental Setup. We instantiate a stratified mixnet topology with $L = 3$ layers and use two datasets to configure our mixnet instances. First, we utilize data from the deployed Nym mixnet to extract inter-node link latencies, node geographic locations (for jurisdiction-aware evaluation), and node reliability, based on historical connectivity to other mixnodes, leveraging the VerLoc protocol [15]. This configuration enables the simulation of mixnet instances with $W = 80$ mixnodes per layer. Additionally, to evaluate DPR at larger scales, we use the RIPE Atlas dataset [12], which provides globally distributed vantage points and empirical latency measurements. Based on this dataset, we simulate mixnets with up to $W = 200$ mixnodes per layer.[10]

Evaluation Metrics. While DPR inherently quantifies anonymity via the differential privacy parameter ε, it is standard in the mixnet literature to assess anonymity using entropy-based metrics. These metrics also enable direct comparison with prior work. To this end, we consider the *routing entropy*, denoted $\mathsf{H}(r)$, originally introduced in [10] and used in follow-up work [11], [19], [20]. This metric is based on Shannon entropy [13] and captures the uncertainty a GPA faces when attempting to infer a message's exit node, given knowledge of its first-hop mixnode.[11]

To evaluate routing optimality or security, we use metrics specific to each optimization objective. For latency optimization, we compute the average end-to-end latency across all selected paths, based on the path PMF and their associated latencies. For unreliability avoidance, we compute the average reliability of selected paths, while for jurisdiction-aware routing, we compute the average number of distinct jurisdictions traversed along each path. We denote these three metrics respectively as ℓ (average latency), $\mathcal{R}_s$ (average reliability), and $\mathcal{J}_n$ (average jurisdictional diversity).

Experimental Results. We begin presenting the experimental results in Fig. 5, which include the average link latency for latency optimization, the optimality of selected paths in terms of unreliability avoidance, and the average number of jurisdictions traversed through the mixnodes when DPR is instantiated for these purposes. The evaluations consider both the Nym and RIPE datasets, and for generality, we examine two configurations: applying the DPR instantiation in isolation, which we refer to as the *vanilla* setting, and applying DPR in conjunction with the TAM algorithm. Results are reported as a function of the privacy parameter ε.[12]

Specifically, Fig. 5a and Fig. 5b present the average link latency under DPR for latency optimization. Fig. 5a corresponds to latency optimization from the client to the last hop in the mixnet—referred to as the *Client-Aware (CA)* setting—while Fig. 5b represents full end-to-end latency optimization from client to destination, which we refer to as the *Client-Destination-Aware (CDA)* setting. Both figures show that latency performs worst around $\varepsilon = 0$, as route selection is performed uniformly at random, increasing the likelihood of selecting poor-quality links. As ε increases, link latency consistently decreases across all scenarios. The reduction is most pronounced for $\varepsilon \leq 2$, after which improvements begin to plateau.

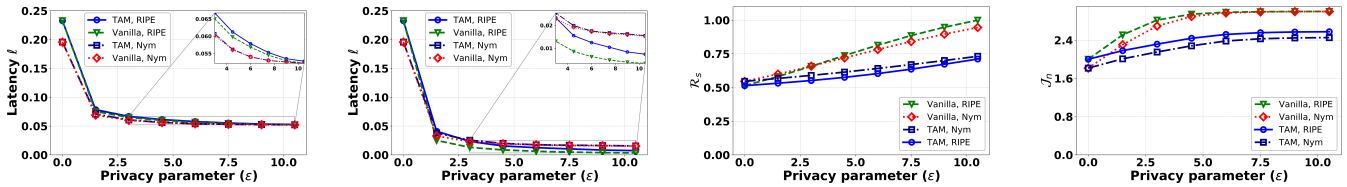
Moreover, across both Nym and RIPE datasets, in both vanilla and TAM settings, CDA provides up to 50 ms lower latency than CA due to the additional optimization from the last mixnode to the destination. Additionally, the RIPE dataset includes a larger and more diverse set of nodes than Nym, encompassing a greater number of nodes with both high- and low-quality links. Consequently, when $\varepsilon = 0$ or under CA, RIPE exhibits higher latency due to a greater likelihood of selecting suboptimal links. However, under higher ε values and with CDA, RIPE achieves greater latency reduction than Nym due to the availability of more high-quality connections. Furthermore, we observe that applying TAM may slightly increase latency compared to the vanilla case. This is because TAM's gradient-based updates adjust the probabilities of forwarding traffic to the succeeding nodes to better align with a balanced target distribution, which may slightly reduce the bias toward highly optimal paths, thereby softening latency improvements.

In addition, ε introduces a trade-off between anonymity and utility. To strike a fair balance between privacy and performance, a reasonable choice of ε is around 3, as it is low enough to maintain a meaningful differential privacy guarantee while still significantly improving latency. Specifically, in this setting, for the RIPE dataset (in the vanilla case), latency is reduced to 65 ms and 15 ms under the CA and CDA strategies, respectively—corresponding to 74% and 94% reductions. For Nym, latency under the same

10. Note that the network size is determined based on the deployed Nym mixnet, and we use $W = 80$ whenever Nym data is used and $W = 200$ when RIPE data is used throughout the paper.

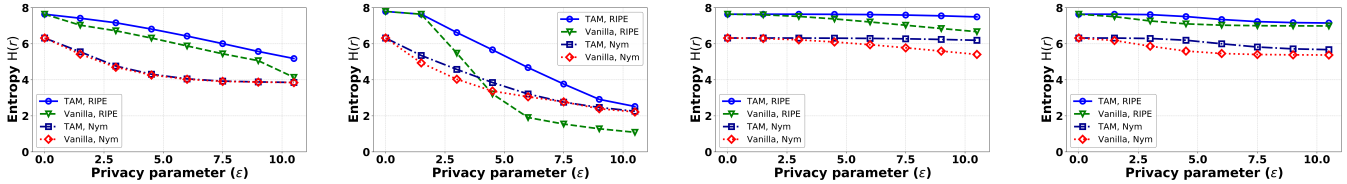
11. To compute $\mathsf{H}(r)$, we first multiply the transition probability matrices P^m for $0 \leq m \leq L-1$. The resulting matrix maps each client to the mixnodes in the final layer, forming a probability distribution over exit nodes. The Shannon entropy of this distribution yields the routing entropy $\mathsf{H}(r)$.

12. Note that our evaluation is limited to the average values in this section. That said, the variance of the resulting data is so low that the averaged results remain reliable.



(a) Latency, CA (sec) (b) Latency, CDA (sec) (c) Reliability score (d) Number of jurisdiction

Figure 5: Analysis of optimality using DPR and TAM: Fig. 5a and Fig. 5b present the results for latency optimization in two scenarios. Fig. 5c and Fig. 5d report the average reliability scores and the average number of jurisdictions traversed, respectively.



(a) Latency optimization, CA (b) Latency optimization, CDA (c) Unreliability avoidance (d) Jurisdiction-aware

Figure 6: Analysis of routing entropy $\mathsf{H}(r)$ when applying DPR and TAM: Fig. 6a and Fig. 6b present the results for latency optimization in two scenarios. Fig. 6c and Fig. 6d present the entropy results for unreliability avoidance and jurisdiction-aware routing instantiations, respectively.

setting is reduced to 60 ms and 22 ms for CA and CDA, respectively, corresponding to 70% and 90% improvements compared to the baseline ($\varepsilon = 0$).

On the other hand, Fig. 5c and Fig. 5d report the average reliability and the number of distinct jurisdictions traversed, respectively, as a function of increasing ε. Across both Nym and RIPE datasets, and under both vanilla and TAM settings, higher values of ε consistently improve both metrics. However, the vanilla setting consistently yields higher reliability scores and a greater number of jurisdictions than the TAM setting. In addition, we observe consistent trends across both datasets, with RIPE showing slightly better average scores in both instantiations overall—likely due to its broader and more diverse node set. Overall, even a moderate increase in ε leads to notable improvements in both reliability and jurisdictional diversity.

Fig. 6 additionally describes the anonymity provided by the mixnet when applying DPR instantiations (both in isolation and together with TAM) for latency optimization, as shown in Fig. 6a and Fig. 6b, unreliability avoidance in Fig. 6c, and jurisdiction-aware routing in Fig. 6d. All measurements are based on the entropy metric $\mathsf{H}(r)$. At first glance, it is evident from Fig. 6 that increasing ε leads to decreasing entropy across all instantiations and their corresponding settings. This behavior is expected, as ε controls the privacy level—higher values reduce randomness in route selection (favoring highly optimal paths), thereby lowering overall anonymity.

Generally, across all settings, applying the TAM algorithm results in higher entropy compared to the vanilla setting. This is because TAM redistributes traffic to succeeding nodes such that the received traffic better aligns with the target distribution. This effect is more pronounced for the RIPE dataset, especially in latency optimization. In particular, compared to Nym, RIPE consistently exhibits higher entropy under both unreliability avoidance and jurisdiction-aware routing. This is due to RIPE having a greater number of nodes, which increases the uncertainty for a GPA attempting to map input nodes to outputs—thus contributing to higher entropy. Overall, both unreliability avoidance and jurisdiction-aware routing maintain high levels of entropy across all ε values (compromising at most 1-bit anonymity), even under high ε, demonstrating that these instantiations preserve anonymity effectively.

Additionally, in the case of CA for latency optimization, we consistently observe higher anonymity compared to CDA. This is because CA introduces greater randomness in route selection, as it only considers the client's links and disregards the destination—thus enhancing anonymity. Moreover, since the RIPE dataset includes a larger number of nodes and contains both lower- and higher-quality links, it preserves more routing randomness in less optimized scenarios (e.g., $\varepsilon \approx 0$ or under CA), resulting in higher anonymity relative to Nym. However, under the CDA strategy, entropy tends to be lower in RIPE as ε increases. This is due to RIPE's higher-quality nodes being selected more frequently, thereby reducing randomness in routing and lowering entropy. That said, when applying the TAM algorithm, the entropy in the RIPE dataset consistently remains higher than that of Nym in both CA and CDA scenarios. Besides this, note that under CDA, for some values of ε, the entropy is as low as 2 bits. In these cases, entropy reduction can be moderated by applying the TAM approach, which increases the likelihood of selecting suboptimal paths and thereby

yields more than 2 bits of entropy.

Lastly, in latency optimization, for $\varepsilon \approx 3$—a reasonable setting for balancing differential privacy—the anonymity for the RIPE dataset is 6.8 and 5.8 bits for CA and CDA strategies in vanilla settings, and up to 7.2 and 6.8 bits in TAM settings, respectively. For Nym, the corresponding values are 4.8 and 4 bits, with TAM producing very similar results. This demonstrates that substantial latency optimization can be achieved at this setting while sacrificing very little in terms of anonymity.[13]

5. Mix-Node Adversary

In this section, we explore an advanced threat model involving an adversary capable of compromising a subset of the mixnodes in the mixnet, referred to as a *mixnode adversary*. Unlike a GPA, which can only passively observe communication links across the network in an attempt to probabilistically correlate input messages with their output counterparts, a mixnode adversary can control specific mixnodes—at which the exact mapping between inputs and outputs is known to the adversary—potentially leading to full deanonymization of the traffic that passes through them. Considering such threats, in this section we assess the robustness of DPR using two metrics. The **first** is an analytical metric known as the *Fraction of Fully Corrupted Paths* (FCP), which measures the proportion of end-to-end message paths (from the first to the last layer of the mixnet) that are entirely compromised—i.e., consist solely of adversarially controlled mixnodes. However, while FCP captures full-path compromise, it fails to account for partial compromises. To address this limitation, we consider the **second** metric—a simulation-based measure of *anonymity of messages*, denoted as $\mathsf{H}(m)$. To this end, we simulate mixnet behavior using the `SimPy` [22] discrete event simulation library in `Python`. We use both the Nym and RIPE datasets to instantiate the network, with a set of clients and destinations. In the simulation, clients generate messages according to a Poisson process at a rate of 30,000 messages per second, forming message paths based on the PMF induced by the DPR-based routing scheme.

In simulations, the mixnodes perform shuffling for unlinkability purposes using the stop-and-go mixing strategy [7], where each message, upon entering any mixnode, experiences a random delay drawn from an exponential distribution with a mean of 50 ms (as in the deployed Nym mixnet). Under this setting, we measure message anonymity by sampling a group of messages at entry and analyzing their posterior distribution over all possible outgoing messages at the final layer, under the assumption that compromised mixnodes offer no anonymity. The entropy of this posterior distribution yields the message anonymity $\mathsf{H}(m)$, following prior works [23], [24]. Unlike FCP, this metric captures both partial and full path compromises and reflects the joint effects of multihop routing and the shuffling process in terms of anonymity.

Having defined the metrics, we further consider two generic adversarial strategies for corrupting mixnodes within the mixnet that reflect potential attack surfaces in real-world deployments of DPR, as follows: *(1) Random Corruption:* Random corruption represents the most straightforward strategy. Here, the adversary controls a fixed number C of mixnodes, selected uniformly at random from the pool of available mixnodes for mixnet construction (see Appendix A, Algorithm 2). *(2) Greedy Corruption:* Greedy corruption represents a more advanced and strategic threat model. In this setting, the adversary computes all possible paths and strategically selects C mixnodes to maximize their overlap with the most frequently used paths. This typically involves ranking nodes by their participation in high-probability routes and selecting those that appear most frequently (see Appendix A, Algorithm 3).

5.1. Experimental Results

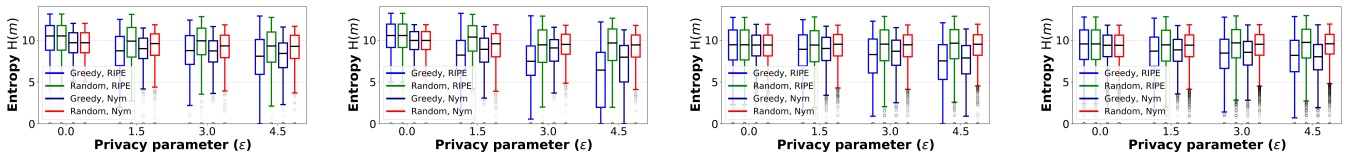
Simulation-Based Evaluation. The first experiment evaluates DPR under the mixnode adversary assumption by simulating the anonymity of messages, measured as entropy $\mathsf{H}(m)$. This analysis considers both the RIPE and Nym datasets, using all DPR instantiations in isolation.[14] The adversary model adopts both the Random and Greedy corruption strategies, assuming that 30% of the mixnodes are compromised. The results are presented in Fig. 7 as a function of the privacy parameter ε. In this figure, the boxplot illustrates the 25th to 75th percentile values, with whiskers extending from the 10th to 90th percentile. Outliers are shown as dark circles.

As Fig. 7 suggests, across all instantiations, increasing ε consistently leads to a decrease in overall message anonymity. This trend arises from two primary effects. First, higher values of ε introduce a stronger bias toward high-utility paths, thereby reducing the randomness in route selection and, consequently, diminishing anonymity. Second, as optimal paths are selected more frequently with larger ε, they are more likely to be compromised by the adversary—particularly under the Greedy corruption strategy. On the other hand, both the RIPE and Nym datasets exhibit similar trends across all instantiations and values of ε. While their average anonymity levels are comparable, the RIPE dataset shows notably higher variance. It includes samples with greater entropy than the best cases observed in Nym, as well as samples with lower entropy. This behavior reflects RIPE's broader mixnode set—comprising a larger number of both high- and low-quality nodes—which increases the likelihood of routing through either highly optimal or suboptimal paths, thereby amplifying the variance in anonymity.

Moreover, as expected, the Greedy corruption strategy leads to greater reductions in message anonymity compared

13. Also note that, inherently, optimizing mixnet performance entails a reduction in anonymity, as formalized by the anonymity trilemma [21]. This effect can be moderated by appropriately setting the privacy parameter ε and by incorporating routing entropy and performance quantifications.

14. DPR instantiations accompanied by TAM yield similar results. For brevity, we report only results under DPR in isolation.



(a) Latency optimization, CA (b) Latency optimization, CDA (c) Reliability score (d) Number of jurisdiction

Figure 7: Simulation-based evaluation of mixnode adversary based on message entropy ($\mathsf{H}(m)$) across different DPR instantiations, with the adversar budget fixed at $\frac{C}{N} = 0.3$.

to Random corruption. This is because Greedy corruption explicitly targets mixnodes that appear most frequently in high-probability paths, whereas Random corruption selects nodes uniformly, often missing the most influential ones.

Furthermore, Fig. 7a and Fig. 7b illustrate entropy results for the latency optimization instantiations under the CA and CDA settings, respectively. CDA consistently results in slightly lower anonymity than CA, as it optimizes both the client-to-mixnet and mixnet-to-destination segments—thereby reducing route randomness further. Nonetheless, both configurations maintain strong anonymity, retaining on average at least 7 bits of entropy even under adversarial compromise.

Finally, Fig. 7c and Fig. 7d present results for unreliability avoidance and jurisdiction-aware routing, respectively. These instantiations also exhibit high resilience against both corruption strategies, consistently maintaining, on average at least over 7.5 bits of message entropy—highlighting the robustness of DPR-based routing under adversarial mixnet scenarios.

Analytical Evaluation (FCP). We next present the analytical results for assessing DPR-based instantiations, shown in Fig. 8, which illustrates the FCP as a function of the privacy parameter ε, with the adversary budget fixed at $\frac{C}{N} = 0.3$. Results are provided for DPR instantiated in the Vanilla setting as well as in conjunction with the TAM algorithm. We evaluate several proposed instantiations, including latency optimization—under both the CA and CDA strategies (Figs. 8a and 8b, respectively)—as well as unreliability avoidance (Fig. 8c) and jurisdiction-aware routing (Fig. 8d). All evaluations assume the Greedy corruption strategy and include results for both the Nym and RIPE datasets.

As shown in Fig. 8, across all instantiations and configurations, FCP consistently increases with larger values of ε. This trend is expected, as higher values of ε introduce stronger biases toward high-utility paths, thereby reducing the randomness in route selection and increasing the likelihood that a mixnode adversary controls frequently selected paths. Additionally, both the Nym and RIPE datasets exhibit similar behavior across all settings and instantiations. However, RIPE—owing to its larger node set—demonstrates slightly improved resilience. Moreover, the results for DPR+TAM, compared to the vanilla setting, consistently show higher resilience against mixnode adversaries. This improvement stems from TAM's mechanism of redistributing the selection probabilities, thereby introducing higher randomness into routing and subsequently reducing the FCP.

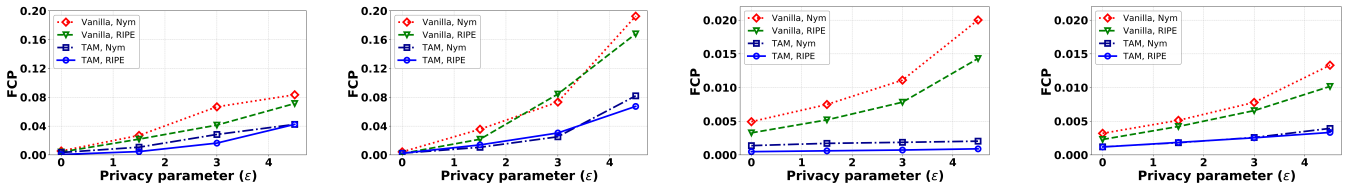
Among the latency optimization settings, CA consistently demonstrates stronger resilience against mixnode adversaries than CDA across both datasets. This is attributed to CA limiting optimization to the client-side segment, thereby preserving greater randomness in routing. Nonetheless, both CA and CDA maintain bounded exposure. Specifically, around $\varepsilon \approx 3$, CA yields an FCP of approximately 0.08 for RIPE and 0.04 for Nym, whereas CDA achieves FCP values of roughly 0.08 for both datasets.

In contrast, Fig. 8c and Fig. 8d report FCP values for the unreliability avoidance and jurisdiction-aware routing instantiations, respectively.[15] In both cases, the observed FCP remains consistently lower—staying below 0.02 and 0.015, respectively—across all ε values, datasets, and settings. These results indicate that DPR-based routing offers strong robustness under adversarial compromise.

Adversarial Budget Evaluation. Fig. 9 presents the FCP as a function of the adversary's corruption budget, expressed by the $\frac{C}{N}$, with $\varepsilon = 3$. Results are reported for DPR in isolation (Vanilla setting) and in conjunction with the TAM algorithm. We evaluate the aforementioned instantiations: latency optimization (Figs. 9a and 9b), unreliability avoidance (Fig. 9c), and jurisdiction-aware routing (Fig. 9d), considering both the Nym and RIPE datasets under the Greedy corruption strategy.

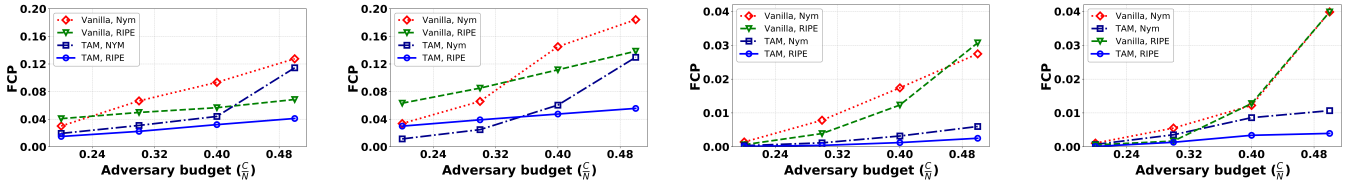
As shown in Fig. 9, across all instantiations and datasets, increasing $\frac{C}{N}$ consistently results in higher FCP values. This trend is expected, as a larger adversarial budget enables the compromise of more mixnodes, thereby increasing the likelihood of fully compromised paths. Additionally, consistent with earlier observations, CDA yields slightly higher FCP values than CA, due to its broader optimization scope, which reduces path selection randomness. Moreover, the unreliability avoidance and jurisdiction-aware routing instantiations show significantly lower FCP values across all corruption levels when compared to latency-optimized strategies. Specifically, for unreliability avoidance, we observe that as the adversarial budget increases, the growth in FCP progresses more slowly compared to that of jurisdiction-aware routing. However, in both cases, the FCP remains bounded—below 0.03 for unreliability avoidance and 0.04 for jurisdiction-aware routing—underscoring the stronger robustness of these approaches under adversarial pressure.

15. The y-axis scales in Fig. 8c and Fig. 8d differ from those in Fig. 8a and Fig. 8b.



(a) Latency optimization, CA (b) Latency optimization, CDA (c) Reliability score (d) Number of jurisdiction

Figure 8: Analytical evaluation of mixnode adversary based on FCP across different DPR instantiations, with the adversary budget fixed at $\frac{C}{N} = 0.3$.



(a) Latency optimization, CA (b) Latency optimization, CDA (c) Reliability score (d) Number of jurisdiction

Figure 9: Analytical evaluation of mixnode adversary based on FCP across different DPR instantiations, with the privacy parameter fixed at $\varepsilon = 3$.

TABLE 1: Comparison of DPR (instantiated for latency optimization) with methods introduced in LARMix [10] and LAMP [11]. Gain is computed as entropy divided by latency (bits/s).

Approach	Latency (ms)		Entropy (bits)		Gain (bits/s)		FCP	
Dataset	Nym	RIPE	Nym	RIPE	Nym	RIPE	Nym	RIPE
Baseline [4]	206	244	6.3	7.7	31	32	0.02	0.02
LARMix [10]	158	171	5.2	6.5	33	38	0.06	0.05
LAMP, SC [11]	74	78	2.9	3.6	39	46	0.13	0.15
LAMP, MC [11]	70	73	2.8	3.6	40	49	0.13	0.15
LAMP, RM, EU [11]	74	79	3.1	3.6	42	46	0.08	0.07
LAMP, RM, NA [11]	91	123	2.6	3.0	29	24	0.03	0.04
DPR, CA	60	65	4.8	6.8	80	105	0.04	0.08
DPR, CDA	22	15	4.0	5.8	182	400	0.08	0.08

6. Discussion

In this section, we present additional experiments evaluating the instantiation of DPR for latency optimization, and compare the results with state-of-the-art approaches, namely LARMix [10] and LAMP [11]. As stated earlier, one of the primary motivations for introducing strategic routing in mixnets is to overcome the high latency incurred due to multi-hop message traversal. Prior prominent efforts addressing this challenge include LARMix [10] and LAMP [11]. Specifically, LARMix proposes a heuristic-based routing mechanism (see LARMix, p. 5), which selects nodes sequentially from the first to the last mixnode along a message route. However, it does not support latency optimization either from the client to the mixnet or from the mixnet to the destination. LAMP, by contrast, attempts to address this limitation by introducing novel mechanisms—Single Circle (SC), Multiple Circles (MC), and Regional Mixnets (RM) (see LAMP, pp. 4–6)—which incorporate latency optimization from the client to the last hop of the mixnet. Nonetheless, it still lacks support for full end-to-end optimization (i.e., including the mixnet-to-destination segment) due to its heuristic framework. In contrast, our DPR-based approach is general, modular, and principled. It supports arbitrary optimization goals (including latency) and allows for optimization both from client to mixnet and from mixnet to destination, all while providing formal anonymity guarantees via differential privacy.

To provide a fair comparison with existing approaches, we adopt the evaluation metrics introduced in § 4—which are also considered in the LARMix and LAMP evaluations—applied to both the Nym and RIPE datasets. These include average latency (ℓ), entropy ($\mathsf{H}(r)$), and the FCP under the Greedy corruption model. We evaluate both CA and CDA instantiations of DPR against LARMix and all variants of LAMP, using the best-performing parameters reported in their respective works. For DPR, we fix $\varepsilon = 3$, focusing on DPR alone (without TAM). As a baseline, we also include the strategy deployed in Nym [4], where intermediate hops are selected uniformly at random. The comparative results are summarized in Tab. 1.

As shown in Tab. 1, when employing the *Baseline* routing strategy, the resulting latency in the mixnet is significantly high, although it provides the highest entropy (i.e., anonymity). To better capture the trade-off between performance and privacy, we define the metric *Gain* as the ratio $\frac{\text{Entropy}}{\text{Latency}}$. Using this metric, the Baseline strategy yields a relatively low gain, making it less practical—despite its strong resilience against mixnode adversaries, as reflected by its minimal FCP under the greedy corruption model. LARMix, on the other hand, slightly reduces latency but at the cost of approximately 1 bit of entropy loss across both the Nym and RIPE datasets. Although it maintains adversarial robustness comparable to the Baseline, its overall performance remains suboptimal due to its limited latency

improvement and reduced anonymity. LAMP's SC and MC variants additionally achieve significantly better latency reduction—bringing latency down to nearly one-third of the Baseline and offering up to $2\times$ improvement over LARMix. However, this latency gain comes with a steep anonymity cost, nearly halving the entropy. As a result, while their Gain metric improves modestly over Baseline and LARMix, they also make the system more vulnerable to mixnode adversaries.

On the other hand, the RM variants of LAMP, evaluated separately for Europe (EU) and North America (NA), exhibit improved robustness against mixnode corruption and moderately better anonymity compared to SC and MC. Specifically, RM (EU) delivers latency similar to SC and MC with slightly higher entropy and comparable Gain. However, RM (NA) performs poorly in all dimensions—offering neither strong anonymity nor competitive latency—resulting in a Gain even lower than that of the Baseline. In contrast, DPR instantiated for latency optimization using the CA strategy clearly outperforms all state-of-the-art methods. CA not only yields the lowest latency among all approaches, but also preserves higher anonymity—surpassing LARMix —and demonstrates strong adversarial robustness. Notably, it achieves over $2\times$ higher Gain than the best-performing alternative.

While CA already offers a strong balance, it does not optimize latency beyond the last mixnode. In contrast, the CDA variant jointly considers both client-to-mixnet and mixnet-to-destination paths. As a result, this strategy achieves latency reductions of 94% and 90% for RIPE and Nym, respectively, compared to the Baseline. At the same time, it maintains higher anonymity than any LAMP variant and delivers a Gain up to $4\times$ that of CA—and more than $8\times$ higher than the best-performing prior method. These results clearly demonstrate the effectiveness of our DPR-based approach in achieving superior performance across all key evaluation metrics.

7. Related Work

Related work to DP-Mix includes: (1) prior efforts that apply differential privacy within mixnets, and (2) research that investigates strategic routing in mixnets.

Differential Privacy in Mixnets. To the best of our knowledge, DP-Mix is the first work to propose strategic routing in mixnets with a quantifiable anonymity loss under a pure ε-differential privacy guarantee. That said, some prior works have applied differential privacy in the context of mixnets, albeit for different objectives. For instance, Vuvuzela [2] and its horizontally scalable variants, Stadium [25] and Karaoke [26], employ differential privacy-inspired techniques to obscure communication patterns by injecting dummy messages that are indistinguishable from real ones. In these systems, if a client (e.g., Alice) communicates with a recipient (e.g., Bob), she may simultaneously send dummy messages to other recipients (e.g., Charlie) to confuse a GPA observing network traffic. These proposals, however, focus on preserving endpoint anonymity through traffic shaping and are not designed to address routing optimization. Moreover, their privacy guarantees are typically formulated under the relaxed (ε, δ)-differential privacy framework, which is a weaker notion compared to the pure ε-differential privacy guarantee achieved by DP-Mix.

Strategic Routing in Mixnets. Strategic routing in mixnets has recently been proposed, but exclusively for latency optimization. Specifically, LARMix [10] introduces a latency-aware routing strategy in which clients select mixnodes that are geographically closer, thereby reducing end-to-end latency. Building on LARMix, LAMP [11] proposes lightweight heuristic-based routing techniques that further reduce latency using circular or regional node selection methods. LARMix and LAMP, however, focus exclusively on latency reduction and rely on heuristic anonymity measures without providing formally quantifiable anonymity, notably because their approaches do not satisfy ε-differential privacy. In contrast, DP-Mix supports a broader range of routing objectives while ensuring formal anonymity guarantees under pure ε-differential privacy. Furthermore, DP-Mix can be extended to enable end-to-end route optimization—from client to destination—unlike LARMix and LAMP, which are not designed to support such comprehensive routing strategies.

Apart from LARMix and LAMP, recent work [27], [28] attempted to generalize the idea of latency optimization in mixnets to broader network settings with formal security guarantees for node arrangements, further introducing innovative cover-traffic generation methods to mask bias in the mixnet caused by low-latency routing. Prior to this, CLAM [20] and LARMix++ [19] extended LARMix to support client-to-mixnet routing and Free Route topologies, while [29]–[31] explored different scenarios of anonymity and latency optimization. That said, all of these approaches are based on heuristic formulas for path selection rather than differential privacy, and additionally focus only on latency improvements.

8. Conclusion

DP-Mix provided a general framework for strategic routing within mixnets, applicable to arbitrary path lengths and customizable for a wide range of routing objectives. It is grounded in differential privacy, offering formal privacy guarantees. Our evaluation demonstrated the framework's flexibility in balancing utility–anonymity trade-offs while preserving strong resilience against mixnode adversaries. The results revealed significant improvements over prior work in latency optimization, positioning DP-Mix as a practical candidate for deployment in real-world mixnets—and potentially in other anonymous communication systems.

Acknowledgments

We thank the anonymous reviewers for their insightful comments, and Russell W. F. Lai for suggesting DP routing in mixnets during IH&MMSec 2024. This research was supported in part by CyberSecurity Research Flanders under reference number VOEWICS02.

References

[1] A. Kwon, D. Lu, and S. Devadas, "XRD: Scalable messaging system with cryptographic privacy," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 759–776.

[2] J. Van Den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, "Vuvuzela: Scalable private messaging resistant to traffic analysis," in *Proceedings of the 25th Symposium on Operating Systems Principles*, 2015, pp. 137–152.

[3] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis, "The loopix anonymity system," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1199–1216.

[4] C. Diaz, H. Halpin, and A. Kiayias, "The Nym network," 2021.

[5] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[6] C. Diaz, "Anonymity and privacy in electronic services," *Heverlee: Katholieke Universiteit Leuven. Faculteit Ingenieurswetenschappen*, 2005.

[7] D. Kesdogan, J. Egner, and R. Büschkes, "Stop-and-go-mixes providing probabilistic anonymity in an open system," in *International Workshop on Information Hiding*. Springer, 1998, pp. 83–98.

[8] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 2007, pp. 94–103.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer, 2006, pp. 265–284.

[10] M. Rahimi, P. K. Sharma, and C. Diaz, "LARMix: Latency-aware routing in mix networks," in *The Network and Distributed System Security Symposium*. Internet Society, 2024.

[11] ——, "LAMP: Lightweight approaches for latency minimization in mixnets with practical deployment considerations," in *The Network and Distributed System Security Symposium*. Internet Society, 2025.

[12] R. N. Staff, "RIPE Atlas: A global internet measurement network," *Internet Protocol Journal*, vol. 18, no. 3, pp. 2–26, 2015.

[13] C. E. Shannon, "Communication theory of secrecy systems," *The Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[14] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *2010 IEEE 51st annual symposium on foundations of computer science*. IEEE, 2010, pp. 51–60.

[15] K. Kohls and C. Diaz, "VerLoc: Verifiable localization in decentralized systems," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2637–2654.

[16] R. Dingledine, N. Mathewson, P. F. Syverson *et al.*, "Tor: The second-generation onion router." in *USENIX security symposium*, vol. 4, 2004, pp. 303–320.

[17] S. Bojja Venkatakrishnan, G. Fanti, and P. Viswanath, "Dandelion: Redesigning the bitcoin network for anonymity," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 1–34, 2017.

[18] G. Fanti, S. B. Venkatakrishnan, S. Bakshi, B. Denby, S. Bhargava, A. Miller, and P. Viswanath, "Dandelion++ lightweight cryptocurrency networking with formal anonymity guarantees," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, pp. 1–35, 2018.

[19] M. Rahimi, "LARMix++: Latency-aware routing in mix networks with free routes topology," in *International Conference on Cryptology and Network Security*. Springer, 2024, pp. 187–211.

[20] ——, "CLAM: client-aware routing in mix networks," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2024, Baiona, Spain, June 24-26, 2024*, F. Pérez-González, P. C. Alfaro, C. Krätzer, and H. V. Zhao, Eds. ACM, 2024, pp. 199–209. [Online]. Available: https://doi.org/10.1145/3658664.3659631

[21] D. Das, S. Meiser, E. Mohammadi, and A. Kate, "Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency-choose two," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 108–126.

[22] Python, "Event discrete, process based simulation for Python." https://pypi.org/project/simpy/, 2013.

[23] I. Ben Guirat, D. Gosain, and C. Diaz, "Mixim: Mixnet design decisions and empirical evaluation," in *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*, 2021, pp. 33–37.

[24] A. M. Piotrowska, "Studying the anonymity trilemma with a discrete-event mix network simulator," in *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*, 2021, pp. 39–44.

[25] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich, "Stadium: A distributed metadata-private messaging system," in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 423–440.

[26] D. Lazar, Y. Gilad, and N. Zeldovich, "Karaoke: Distributed private messaging immune to passive traffic analysis," in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, 2018, pp. 711–725.

[27] M. Rahimi, "OptiMix: Scalable and distributed approaches for latency optimization in modern mixnets," in *The Network and Distributed System Security Symposium*. Internet Society, 2026.

[28] ——, "OptiMix: Scalable and distributed approaches for latency optimization in modern mixnets (extended version)," *Cryptology ePrint Archive*, 2026.

[29] ——, "PARSAN-Mix: Packet-aware routing and shuffling with additional noise for latency optimization in mix networks," in *International Conference on Applied Cryptography and Network Security*. Springer, 2025, pp. 159–188.

[30] ——, "Malaria: management of low-latency routing impact on mix network anonymity," in *2024 22nd International Symposium on Network Computing and Applications (NCA)*. IEEE, 2024, pp. 193–202.

[31] ——, "MOCHA: Mixnet optimization considering honest client anonymity," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2025, pp. 98–107.

[32] Y. Sun, A. Edmundson, N. Feamster, M. Chiang, and P. Mittal, "Counter-RAPTOR: Safeguarding tor against active routing attacks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 977–992.

[33] C. Diaz and B. Preneel, "Taxonomy of mixes and dummy traffic," in *Information Security Management, Education and Privacy: IFIP 18th World Computer Congress TC11 19th International Information Security Workshops 22–27 August 2004 Toulouse, France*. Springer, 2004, pp. 217–232.

[34] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman, "Mixmaster protocol—version 2," 2003.

Appendix

A. Algorithms

In this section, we present the pseudocode for the TAM in Algorithm 1. Additionally, the *Random Corruption* and *Greedy Corruption* algorithms, which model the capabilities of a mixnode adversary, are provided in Algorithm 2 and Algorithm 3, respectively.

Algorithm 1 Traffic Alignment Mechanism (TAM)

Input: Transition probability matrices P^m and target distribution vectors q^{m+1} for $0 \leq m \leq L-1$, stopping criterion δ, and learning rate λ.
Define

$$\text{Cost}(m) = \text{tr}\left[\left((P^m)^\top \mathbb{1} - q^{m+1}\right)\left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right].$$

$$\text{Gradient: } \exp\left[\text{tr}\left((S_{ij}^m)^\top \mathbb{1} \cdot \left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right)\right].$$

for each $m \in \{0, 1, \ldots, L-1\}$ **do**
  while Cost$(m) > \delta$ **do**
    for each entry $p_{ij}^m \in P^m$ **do**
      $p_{ij}^m \leftarrow p_{ij}^m - \lambda \cdot$ Gradient
    Recompute Cost(m)
Output: All updated matrices P^m for $m \in \{0, \ldots, L-1\}$

Algorithm 2 Random Corruption

Input: Set of available mixnodes $\mathcal{S}_M$, budget C
Initialize $\mathcal{S}_A \leftarrow \emptyset$ ▷ Set of selected/adversarial mixnodes
Set counter $\leftarrow 0$
while counter $< C$ **do**
  Randomly select a mixnode from $\mathcal{S}_M$
  Add the selected node to $\mathcal{S}_A$
  Remove it from $\mathcal{S}_M$
  counter $\leftarrow$ counter $+1$
Output: Set of corrupted mixnodes $\mathcal{S}_A$

Algorithm 3 Greedy Corruption

Input: Set of available mixnodes $\mathcal{S}_M$, budget C, prior information $\mathcal{I}$
Initialize $\mathcal{S}_A \leftarrow \emptyset$ and FCP $\leftarrow 0$
From $\mathcal{I}$, compute the set $\mathcal{S}_P$ of all paths in the system
Set counter $\leftarrow 0$
while counter $< C$ **do**
  if counter == 0 **then**
    Select L nodes from $\mathcal{S}_M$ that maximize FCP
  else
    Select 1 node from $\mathcal{S}_M$ that maximizes the increase in FCP
  Add the selected node to $\mathcal{S}_A$
  Remove it from $\mathcal{S}_M$
  Update FCP based on the new set $\mathcal{S}_A$
  counter $\leftarrow$ counter $+1$
Output: Set of corrupted mixnodes $\mathcal{S}_A$

B. Background

In this section, we briefly explain the concept of pure ε-differential privacy and the exponential mechanism, both of which we used as subroutines for strategic routing in mixnets.

Pure ε-Differential Privacy. Consider a scenario where there are n individuals, each with their own data point, denoted by $X_1, X_2, \ldots, X_n$. A randomized algorithm M is applied to the dataset to produce an output that depends on the data in a probabilistic manner. Differential privacy is a property of such an algorithm M, ensuring that no single individual's data has a significant impact on the output. More formally, let $M : \mathcal{X}^n \to \mathcal{Y}$ be a randomized algorithm. For any two datasets $X, X' \in \mathcal{X}^n$ that differ in exactly one entry—called neighboring datasets—M is said to be *pure ε-differentially private* if for all subsets $T \subseteq \mathcal{Y}$, the following inequality holds: $\Pr[M(X) \in T] \leq e^\varepsilon \cdot \Pr[M(X') \in T]$, where the probability is taken over the internal randomness of the algorithm M. This definition guarantees that the inclusion or exclusion of any single individual's data does not substantially affect the probability of any output, thereby preserving privacy [9].

Exponential Mechanism. The exponential mechanism [8] is a fundamental tool in differential privacy, particularly useful when the dataset domain $\mathcal{X}^n$ is non-numerical. In such cases, each data point in $X \in \mathcal{X}^n$ is associated with one of many possible outputs, or objects, from a set $\mathcal{H}$. Among these, there may exist a "best" object $h^* \in \mathcal{H}$, as determined by a scoring function $s : \mathcal{X}^n \times \mathcal{H} \to \mathbb{R}$, where higher scores indicate better utility.

If we were to deterministically select the object with the highest score, we could unintentionally reveal sensitive information about the underlying dataset. To preserve privacy, the exponential mechanism selects objects probabilistically, favoring higher scores while maintaining differential privacy.

Let the sensitivity of the score function be defined as:

$$\Delta_s = \max_{h \in \mathcal{H}} \max_{X, X'} |s(X, h) - s(X', h)|,$$

which measures the maximum change in score for any object h when a single data point in the dataset is changed. Given inputs $X, \mathcal{H}, s$, the exponential mechanism M_E selects and outputs an object $h \in \mathcal{H}$ with probability proportional to: $\Pr[M_E(X) = h] \propto \exp\left(\frac{\varepsilon \cdot s(X,h)}{2\Delta_s}\right)$. As shown in [8], the exponential mechanism satisfies pure ε-differential privacy and provides strong utility guarantees by ensuring that the selected object has a score close to that of the optimal object h^*, with high probability.

C. TAM Algorithm

In this section, we provide a detailed derivation of the gradient descent update rule used in the TAM algorithm, along with its preservation of differential privacy guarantees.

Proof of Lemma 1. We begin by recalling the cost function minimized by TAM:

$$\text{Cost} = \text{tr}\left[\left((P^m)^\top \mathbb{1} - q^{m+1}\right)\left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right].$$

To minimize this cost, TAM applies a gradient descent procedure to iteratively update the entries of P^m. In this derivation, we compute the derivative of the loss with respect to each entry of P^m. To facilitate the computation and

ensure proper coverage of the exponential parameterization, we define $p_{ij}^m = e^{\alpha_{ij}^m}$, and perform the gradient step with respect to α_{ij}^m.

Let us now derive the gradient required to update p_{ij}^m in TAM. The derivative of Cost with respect to α_{ij}^m is computed through the steps below, as labeled by Equations (3)–(6):

$$\frac{\partial \text{Cost}}{\partial \alpha_{ij}^m},$$

$$= \frac{\partial}{\partial \alpha_{ij}^m} \operatorname{tr}\left[\left((P^m)^\top \mathbb{1} - q^{m+1}\right)\left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right], \tag{3}$$

$$= \operatorname{tr}\left[\frac{\partial}{\partial \alpha_{ij}^m}\left((P^m)^\top \mathbb{1} - q^{m+1}\right)\left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right], + \operatorname{tr}\left[\left((P^m)^\top \mathbb{1} - q^{m+1}\right)\frac{\partial}{\partial \alpha_{ij}^m}\left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right], \tag{4}$$

$$= \operatorname{tr}\left[(S_{ij}^m)^\top \mathbb{1} \cdot \mathbb{1}^\top P^m - (S_{ij}^m)^\top \mathbb{1} \cdot (q^{m+1})^\top\right], + \operatorname{tr}\left[(P^m)^\top \mathbb{1} \cdot \mathbb{1}^\top S_{ij}^m - q^{m+1} \cdot \mathbb{1}^\top S_{ij}^m\right], \tag{5}$$

$$= 2 \cdot \operatorname{tr}\left[(S_{ij}^m)^\top \mathbb{1} \cdot \left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right]. \tag{6}$$

Now, α_{ij}^m can be updated via gradient descent as follows:

$$\alpha_{ij}^m \leftarrow \alpha_{ij}^m - \lambda \cdot \operatorname{tr}\left[(S_{ij}^m)^\top \mathbb{1} \cdot \left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right].$$

However, since $p_{ij}^m = e^{\alpha_{ij}^m}$, the corresponding update in terms of p_{ij}^m is:

$$p_{ij}^m \leftarrow p_{ij}^m - \lambda \cdot \exp\left[\operatorname{tr}\left((S_{ij}^m)^\top \mathbb{1} \cdot \left(\mathbb{1}^\top P^m - (q^{m+1})^\top\right)\right)\right].$$

Post-Processing Feature of DPR. In § 2, while introducing TAM, we stated that TAM does not compromise the anonymity guarantees provided by DPR. This stems from a fundamental property of differentially private algorithms: once a quantity is privatized, it cannot be "de-privatized," assuming the underlying data is not accessed again. We formalize this feature of DPR in Lemma 2, showing that not only TAM, but any deterministic or randomized algorithm, cannot de-privatize the outputs of a differentially private DPR mechanism. (The proof is based on the construction presented in [9].)

Lemma 2. *Let $M_{DPR} : \mathcal{X}^n \to \mathcal{P}$ be an ε-differentially private DPR mechanism based on the exponential mechanism, where $\mathcal{X}^n$ is a set of tuples of client–destination pairs, and $\mathcal{P}$ is a set of possible path selection distributions. Let $G : \mathcal{P} \to \mathcal{P}'$ be an arbitrary randomized function (e.g., TAM). Then $G \circ M_{DPR}$ is ε-differentially private.*

Proof. Since G is a randomized function, it can be viewed as a distribution over deterministic functions g. In this case, the privacy proof proceeds as follows for every pair of neighboring datasets X, X' and for every measurable subset $A \subseteq \mathcal{P}$ (representing specific events in the output of M_{DPR}):

$$\begin{aligned}\Pr[G(M_{DPR}(X)) \in A] &= \mathbb{E}_{g\sim G}[\Pr[M_{DPR}(X) \in g^{-1}(A)]],\\ &\le \mathbb{E}_{g\sim G}[e^{\varepsilon}\Pr[M_{DPR}(X') \in g^{-1}(A)]],\\ &= e^{\varepsilon}\Pr[G(M_{DPR}(X')) \in A].\end{aligned}$$

D. More Optimization Purposes

In this section, we outline additional optimization objectives that can be supported under the DPR framework for route selection in mixnets. In particular, we focus on two extended use cases: *Congestion-Aware Routing* and *Joint Latency-and-Congestion-Aware Routing*.

Congestion-Aware Routing Mixnets such as the Nym network are decentralized infrastructures composed of incentivized, volunteer-operated mixnodes distributed globally. These nodes provide mixing services and are compensated based on their participation. However, mixnodes often differ significantly in their bandwidth capacities, leading to variability in their ability to handle traffic. Consequently, routing paths that include low-capacity nodes may experience congestion, increased delays, or message loss—ultimately degrading overall network performance. To address this challenge, we instantiate DPR to support *congestion-aware routing*, which explicitly considers node capacities during path selection. To the best of our knowledge, this is the first approach that enables congestion-aware path design in mixnets and can be directly integrated into real-world systems such as Nym to enhance reliability and throughput.

We consider the dataset $X \in \mathcal{X}^n$ as a set of client-destination tuples and define the set $\mathcal{S}_P^{\mathrm{c}}$ as the collection of all candidate paths for client c, each representing a complete route through the mixnet.[16] In this setting, we define the utility function to prioritize paths that avoid bottlenecks. For client c and path $P_i^{\mathrm{c}} \in \mathcal{S}_P^{\mathrm{c}}$, the congestion-aware utility is given by: $\nu_i^{\mathrm{c}} = \min\left(\omega_{1i}^{\mathrm{c}}, \omega_{2i}^{\mathrm{c}}, \ldots, \omega_{Li}^{\mathrm{c}}\right)$, where ω_{ji}^{c} denotes the bandwidth capacity of the j-th node in path P_i^{c}. This utility captures the intuition that the effective throughput of a path is limited by its most constrained (lowest-capacity) node.

The global score function is then computed using Eq. (1), where the sensitivity Δ reflects the maximum change in utility resulting from a change in a single client-destination pair. Assuming uniform traffic generation, we set $a_{\mathrm{c}} = \frac{\omega^*}{|\mathcal{S}_C|}$, where ω^* denotes the maximum node capacity in the mixnet. Applying the exponential mechanism, each configuration $h_j \in \mathcal{H}$ is selected with probability proportional to: $\exp\left(\frac{\varepsilon \cdot s(X,h_j)}{2\Delta}\right)$, thereby probabilistically favoring paths with greater throughput, while still ensuring pure ε-differential privacy.

Joint Latency-and-Congestion-Aware Routing. Finally, we consider a scenario in which DPR is used to jointly optimize for both low latency and high capacity. This setting

16. As destinations do not influence node capacities, they are not needed for congestion-aware scoring.

reflects practical use cases where performance depends not only on minimizing delay but also on avoiding bandwidth bottlenecks.

In this case, for a given path $P_i^{\subset}$, we define the utility score as: $\nu_i^{\subset} = \frac{\min\{\omega_k^{\subset} | k \in P_i^{\subset}\}}{\ell_i^{\subset}}$, where ω_k denotes the bandwidth capacity of node $k \in P_i^{\subset}$, and $\ell_i^{\subset}$ is the total latency of path $P_i^{\subset}$. This formulation rewards paths that have both high minimum capacity and low overall latency. The corresponding sensitivity of the score function is given by: $\Delta = \frac{\omega^*}{\ell^*}$, where ω^* is the maximum node capacity and ℓ^* is the minimum latency observed across all candidate paths in the network. This joint utility function can be seamlessly integrated into the DPR model using Eq. (1). These examples demonstrate the flexibility of DP-Mix in supporting a broad range of routing objectives beyond those explicitly evaluated in the main body of this work.

E. Application of DP-Mix and Future Work

In this section, we highlight the potential applicability of DPR to other anonymous communication networks.

DP-Mix is a generic and flexible model that can be adapted to a wide range of optimization objectives—not limited to latency or congestion—as discussed earlier.[17] This adaptability makes DP-Mix well-suited for future use cases that require dynamic, utility-driven routing decisions. Furthermore, the general formulation of DP-Mix supports deployment at various granularities, ranging from one-hop segment selection to complete end-to-end path construction. Importantly, the applicability of DP-Mix is not confined to mixnets; it can be extended to other anonymous communication systems. For instance, DP-Mix-based methods could be employed to optimize path selection in Tor [16], with objectives such as minimizing latency, strengthening guard node protection, or improving resilience against traffic correlation attacks. These applications have the potential to outperform state-of-the-art techniques in Tor, including those proposed in [32]. Similarly, DP-Mix can be integrated into I2P[18] to strike a balance between latency and anonymity. On the other hand, recent efforts have focused on enhancing transaction-level anonymity in blockchain systems, as explored in [18]. However, many of these heuristic-based approaches fail to provide strong anonymity guarantees while maintaining high performance. In such scenarios, DP-Mix presents a promising alternative by offering both formal privacy guarantees and optimization flexibility—paving the way for a new line of research into utility-aware routing for anonymous systems.

F. Different Mixnet Designs

In this section, we first introduce the different mixing processes and the topology design of mixnets. We then detail how our design can be extended to other types of mixnets.

The traffic mixing performed by mixnodes, which ensures message unlinkability, can be implemented through various methods. The **first** approach is *threshold mixing* [5], where messages are accumulated until a certain message count is reached before they are forwarded. Alternatively, messages can be held until a specified time interval elapses, a variation known as *timed mixing*. The **second** approach uses *pool mixes* [33], where messages are forwarded only after both a threshold number of messages and a time limit have been reached. The **third** approach, *stop-and-go mixing* [7], treats each message independently, introducing a delay before forwarding. This delay is determined by a delay function that follows an exponential distribution ($\mathsf{Exp}(\mu)$) with parameter μ.

In addition to different mixnode types, mixnet topologies can be structured in various ways. For a mixnet consisting of L intermediary hops, one common configuration is the *cascade* topology, where mixnodes are organized into cascades (mix-chains), each containing L mixnodes. Clients select one of these cascades for message routing [2]. Another configuration, known as *free routes*, allows clients to randomly select L distinct nodes from all available mixnodes to establish a message path [34]. A third approach is the *stratified* topology, which we considered in DP-Mix, where mixnodes are organized into L distinct layers, and a message path is formed by selecting one mixnode from each layer [3], [4].

In this paper, we focused, however, on differential privacy-based routing under the stratified topology of mixnets. That said, our approach can be extended to other topologies such as cascade and free-route configurations. For example, in free-route topologies, the number of possible paths, assuming N available mixnodes, is $N \times (N-1) \times \cdots \times (N-L+1)$ for each client $\subset \in \mathcal{S}_C$. By defining each path $P_i^{\subset} \in \mathcal{S}_P^{\subset}$ as one of the available routes and computing its corresponding score, we can adapt our DPR-based routing accordingly. Similarly, for the cascade topology, we consider each client $\subset \in \mathcal{S}_C$ and define $P_i^{\subset} \in \mathcal{S}_P^{\subset}$, where the cardinality of the path set is $\frac{N}{L}$. In this setting, all valid paths can be constructed with their associated scores, enabling the application of our DPR-based routing method to these topologies as well.

G. Some Clarifications

In §2, we assumed that each client has a single destination to communicate with when performing differentially private routing (DPR). Note that this assumption does not reduce the generality of the approach: in scenarios where a client may have multiple potential destinations, one can simply treat each destination as a separate tuple. This allows the method to remain directly applicable without modification.

On the other hand, in §2, when defining path utility, we primarily described it as being client-specific. However,

17. See additional optimization objectives in Appendix D.
18. https://geti2p.net

these utility values can also be defined in a client-agnostic manner. That is, instead of maintaining a dedicated path set per client, one may consider a global path set $\mathcal{S}_P$ shared across all clients. Each path $P_i \in \mathcal{S}_P$ would then be assigned a utility score ν_i that is independent of the client. In short, designers can benefit from the flexibility of the DPR method and adapt it seamlessly in their own use cases.