When Mixnets Fail: Evaluating, Quantifying, and Mitigating the Impact of Adversarial Nodes in Mix Networks

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

***Abstract*—Mix networks (mixnets) provide clients with communication anonymity against strong network adversaries by traversing their packets independently through randomly selected hops (mixnodes), which disrupt packet linkability. Although this approach, implemented in Nym, maximizes obfuscation against network adversaries, it enables an adversary who compromises a subset of mixnodes (10%/5% of nodes) to entirely nullify the anonymity of all clients whose communication volume with their destination exceeds a certain threshold (4 MB/30 MB).**

To mitigate such vulnerabilities, this work develops a set of novel path selection techniques that achieve a trade-off between resistance to network adversaries and resilience against compromised mixnodes. Observing that existing anonymity metrics are insufficient to quantify adversarial risk in mixnets, we additionally introduce effective empirical and simulation-based metrics. Through theoretical, empirical, and simulation-based evaluations, we comprehensively assess our proposals, demonstrating that the proposed approaches reduce the vulnerability to compromised nodes by up to 80%, while conferring limited advantage to network adversaries. Our analysis further reveals that state-of-the-art anonymity metrics, in contrast to our proposed metrics, produce misleading results that influenced certain design choices in Nym.

I. Introduction

Concealing the relationship between communicating parties on the Internet, mix networks (*mixnets*) [6], [15], [5], [41], [25], [21], [9] stand out as an effective anonymous communication system. Mixnets operate as overlay networks that forward clients' communication packets through multiple intermediaries (*mixnodes*), randomizing their input traffic order to render tracing packets back to their originators (clients) infeasible for a network adversary observing all communication exchanges, commonly referred to as a *Global Passive Adversary* (GPA). Mixnets further differ in the connectivity among their mixnodes (*network topologies*) [10] and in how incoming traffic patterns are transformed within mixnodes (*mixing processes*) [39]. In this work, we focus on a Loopix-like design [25], as it forms the basis of the only mixnet currently deployed in practice—the Nym network [9]. In this design, nodes are arranged into layers such that each node in layer ℓ is only connected to nodes in layers $\ell-1$ and $\ell+1$. Consequently, each packet traverses exactly one node from each layer before reaching its destination. Additionally, for mixing purposes in Loopix, each mixnode flushes incoming packets after a random delay sampled from an exponential distribution [18], a suitable scheme for real-time communications.

Within Loopix (Nym), client traffic is transmitted to its destination by fragmenting the traffic data into fixed-size packets using the Sphinx packet format [8], an efficient scheme that enhances packet unlinkability [25], [9]. Notably, each Sphinx packet carries a small payload (2 KB in Nym[1]) and is routed independently via a path selected uniformly at random, consisting of one node from each mixnet layer. This design ensures that each route consistently carries a comparable number of packets, thereby reducing the GPA's ability to infer communication patterns based on traffic volume [9].

Although the aforementioned design choice effectively thwarts the threat posed by the GPA, its impact on the advantage of *mixnode adversaries*, compromising a subset of nodes within the mixnet, has not been thoroughly examined. Specifically, such adversaries can fully deanonymize clients if any of their packets traverse only adversarial nodes. While the threat of mixnode adversaries has been largely overlooked in state-of-the-art mixnet works [9], [16], [34], [35], their lower infrastructure requirements make them a more practical threat in real-world scenarios.

Design Goals. Given the practical threat of mixnode adversaries, this paper aims to: (1) precisely quantify the advantage gained by such adversaries in the Loopix (Nym) network; (2) introduce practical path assignment strategies designed to mitigate this advantage; (3) propose new evaluation metrics that accurately capture the adversarial threat; (4) conduct a thorough evaluation of our techniques and examine their impact on various aspects of mixnets, exploring whether reducing the advantage of mixnode adversaries can be achieved without significantly increasing the advantage of the GPA.

[1] https://nym.com/blog

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.242384
www.ndss-symposium.org

Our Contributions. Our stated goals are achieved through the following key contributions. **First**, using data from the deployed Nym mixnet and under the Nym packetization format, we quantify the risk imposed by mixnode adversaries on client–destination pair deanonymization. We demonstrate that when an adversary controls 20%, 10%, or 5% of the mixnodes, clients exchanging 1 MB, 4 MB, or 30 MB of data, respectively, are fully deanonymized—that is, at least one of their communication packets traverses a path composed entirely of adversarial nodes.[2]

Second, to moderate the extensive risk of deanonymization posed by mixnode adversaries, we propose approaches that aim to increase the similarity of paths used for packets belonging to a distinct client–destination session. To this end, we introduce three strategies: (1) the *K-Hops Fixed* (K-HF) strategy, (2) the *K/W* strategy, and (3) the α-*Sticky Selection* (α-SS) strategy.

Under the K-HF strategy, assuming a stratified mixnet with L layers, each containing W mixnodes, the path assigned to packets belonging to a client–destination session is constructed by initially fixing a subset of layers (e.g., the subset $\{1, 2\}$ when $L \geq 2$). For each layer in this subset, a single node is selected uniformly at random. All packets in the session are then required to traverse these fixed nodes when passing through the preselected layers, while the nodes in the remaining layers are selected independently and randomly for each packet. By enforcing this partial determinism in path selection, the K-HF strategy reduces path diversity, thereby limiting the chance of any packet traversing a fully compromised path.

While K-HF is effective in limiting the advantage of mixnode adversaries, its reliance on fixed hops for transferring all packets in the session may lead to overloading the preselected nodes, especially under high-traffic regimes. To moderate such effects, we propose the alternative *K/W* strategy. This approach initially selects K mixnodes out of the W available in each layer. Then, to construct paths for each packet, it independently samples one node per layer from the preselected K candidates. This reduces the adversary's overall advantage by limiting the number of possible paths, while lowering the likelihood of node overloading by providing more mixnodes for selection.

The α-SS strategy, on the other hand, takes a probabilistic approach to reduce path diversity rather than explicitly fixing or restricting the path set. Specifically, in α-SS, the path for the first packet in a client–destination communication flow is selected uniformly at random from the set of all available paths. For the second packet, the previously selected path is reused with probability α; with probability $1 - \alpha$, a new path is selected uniformly at random from the unassigned paths. This process continues such that for the i-th packet in the session, with probability α, one of the paths used by any of the previous $i - 1$ packets is reused, and with probability $1 - \alpha$, a new path is assigned at random from the remaining unassigned paths. Depending on the configuration of α, this strategy can moderate both the risk posed by mixnode adversaries and the load distribution across mixnodes.

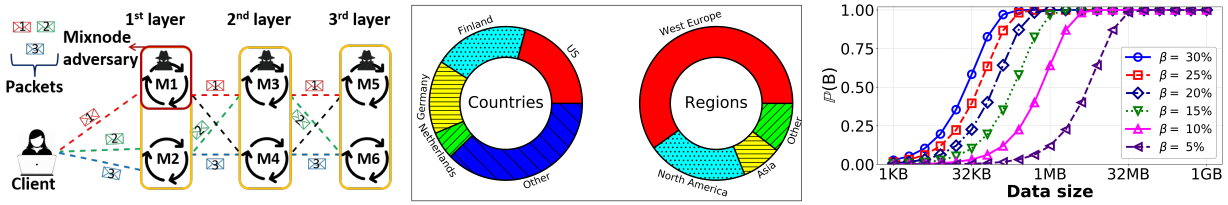
Third, we introduce new evaluation metrics supported by theoretical, empirical, and simulation-based analyses. In the theoretical analysis, we derive probabilistic bounds on the advantage of mixnode adversaries in deanonymizing client–destination sessions, as well as upper bounds on the advantage of the GPA and on the reliability of communications in the presence of unreliable nodes.

These theoretical results, however, rely on a set of default assumptions that limit their accuracy. To more precisely quantify the advantage of both mixnode adversaries and the GPA, we further conduct empirical analyses of mixnets. Specifically, we configure the mixnet with different samples of nodes being compromised in each mixnet layer and assess the advantage of both the mixnode adversary and the GPA under our proposed path selection strategies. To this end, we introduce two empirical metrics: (1) the *Deanonymization Likelihood Metric* (DLM), which quantifies the advantage of the mixnode adversary as the number of deanonymized sessions in which at least one packet traverses a fully compromised path, normalized over all sessions; and (2) the *Correlation Advantage Metric* (CAM), which measures the GPA's ability to distinguish communication patterns between specific client–destination pairs.

Theoretical and empirical scenarios, nonetheless, assess GPA and mixnode adversaries in isolation and do not account for the mixing process performed by mixnodes during packet transmission. To capture the combined effect of both adversaries while accounting for the mixing process, we finally consider simulation scenarios. In this setting, a set of client–destination pairs exchange packets through mixnets, with packet transmissions modeled as discrete-event processes using the `SimPy` [26] environment in `Python`. In simulations, we define a session-based anonymity metric. This metric simulates the joint view of the GPA (observing all communication exchanges) and mixnode adversaries (with knowledge of the input–output mapping at compromised nodes). Based on this joint view, and upon the exit of each packet, it constructs an empirical distribution over the possible client–destination pairs from which the exit packet could have originated. The Shannon entropy [38] of this distribution, denoted by $\mathsf{H}(S)$, serves as our simulation-based anonymity metric.

Fourth, we perform extensive evaluations across our theoretical, empirical, and simulation-based scenarios. Our results show that, compared to the baseline, the advantage of mixnode adversaries is reduced by up to 80%, 85%, and 80% when using the K-HF, K/W, and α-SS strategies, respectively, while the GPA's advantage increases only marginally—by 1.5 bits, 2 bits, and 1 bit, respectively. Simulation results further reveal that state-of-the-art anonymity metrics, in contrast to our proposed metrics, produce misleading results that have influenced certain design choices in Nym—such as the adoption of loop-based cover traffic, which fails to reduce the threat posed by mixnode adversaries.

[2]Even adversaries compromising fewer mixnodes can achieve full deanonymization once a certain data transmission threshold is exceeded.



(a) A mixnet example (b) Mixnodes distributions (c) Probability of deanonymization

Fig. 1: Fig. 1a illustrates a stratified mixnet composed of $L = 3$ layers, each containing $W = 2$ nodes, with a client transmitting three packets through the mixnet. Fig. 1b shows the geographical distribution of mixnodes in the Nym network. Fig. 1c presents the probability of deanonymization as a function of the volume of data transferred by the client.

II. System Model and Methodology

This section details the threat posed by mixnode adversaries and introduces our proposed mitigations along with their theoretical analysis.

A. Mixnode Adversaries Threat

Loopix [25] protects client privacy by forwarding their traffic through a sequence of L layers, each containing W mixnodes. For instance, Fig. 1a illustrates such a mixnet with $L = 3$ and $W = 2$. Specifically, these mixnets are designed to mitigate the threat posed by a GPA capable of observing all communication exchanges over the Internet. A key design principle supporting this goal is that a client's outgoing data is fragmented into small packets, each of which is routed independently, with its path selected uniformly at random from the set of all possible paths. This design ensures that each path within the mixnet carries approximately the same volume of traffic, thereby maximizing uncertainty for the GPA when attempting to distinguish the path selected by clients.

Although this key design choice minimizes the advantage of a GPA, its impact on mixnode adversaries is often reduced to reporting only the fraction of fully compromised paths in mixnets [9], [16], [34]. Under this form of quantification, Fig. 1a shows that only one out of eight possible paths—namely, (M_1, M_3, M_5)—is fully compromised, yielding a fraction of 0.125. However, this approach fails to account for the number of packets transmitted per session. When multiple packets (e.g., three within the same session) are sent, the probability of deanonymization—that is, the likelihood that at least one packet traverses a fully compromised path—increases significantly (e.g., $1 - (1 - 0.125)^3 \approx 0.33$ in this example).

This deanonymization probability is exacerbated as the number of transmitted packets increases. To examine this more formally in the context of Nym, let $\mathcal{S}_L = \{1, 2, \cdots, L\}$ denote the set of mixnet layers, and let $\mathcal{S}_{M_j}$ represent the set of mixnodes in the j-th layer, where $j \in \mathcal{S}_L$. Assume that a client uses the mixnet to transmit a chunk of data to a destination. We denote this data by the set $\mathcal{S}_D = \{p_1, p_2, \cdots, p_m\}$, where each p_i represents the i-th packet in the client–destination communication session, and m is the total number of packets in $\mathcal{S}_D$. In the Nym design, each packet in $\mathcal{S}_D$ is assigned a path independently and uniformly at random. We represent the path assigned to p_i as the sequence $M_1^i, M_2^i, \cdots, M_L^i$, where each M_j^i is drawn uniformly at random from the set $\mathcal{S}_{M_j}$. We denote this path by the set $\mathcal{S}_{p_i}$, formally expressed in Eq. (1).

We further define $\mathcal{S}_C$ as the set of all adversarial mixnodes in the mixnet. Let A denote the event that a particular packet's selected path is fully compromised, and let B denote the event that *at least one* packet in $\mathcal{S}_D$ experiences event A, as expressed in Eq. (2). Under this setting, the probability of deanonymization for the set $\mathcal{S}_D$, denoted by $\mathbb{P}(\mathsf{B})$, is given by Eq. (3), where the product term captures the probability that all m packets are routed through uncompromised paths (i.e., each $\mathcal{S}_{p_i}$ contains at least one honest node). Additionally, assume the adversary controls a fraction β of the total $L \cdot W$ mixnodes in the network, i.e., $|\mathcal{S}_C| = \beta LW$. Depending on how nodes are assigned to layers, typically uniformly at random, the impact of this compromise may vary. In the most likely scenario (which also corresponds to the adversary's best-case), the compromised mixnodes are evenly distributed across all layers, such that a fraction β of nodes in each layer are adversarial. Under this assumption, the probability that a randomly selected path for packet p_i traverses only compromised nodes in every layer, i.e., $\mathbb{P}(\mathsf{A} \mid p_i, \mathcal{S}_{p_i})$, becomes β^L. Substituting this into Eq. (3), we obtain the simplified expression in Eq. (4). This expression reveals that as m increases, the term $(1 - \beta^L)^m$ decays exponentially (since $0 \leq \beta^L \leq 1$), and consequently, the overall deanonymization probability $\mathbb{P}(\mathsf{B})$ increases rapidly with the number of transmitted packets.

$$\mathcal{S}_{p_i} = \left\{ M_1^i, M_2^i, \cdots, M_L^i \,\middle|\, \forall j \in \mathcal{S}_L,\ M_j^i \xleftarrow{\$} \mathcal{S}_{M_j} \right\}.^3 \quad (1)$$

$$\mathsf{B} = \left\{ \exists\, p_i \in \mathcal{S}_D \text{ s.t. } M_j^i \in \mathcal{S}_C \wedge M_j^i \in \mathcal{S}_{p_i},\ \forall j \in \mathcal{S}_L \right\}. \quad (2)$$

$$\mathbb{P}(\mathsf{B}) = 1 - \prod_{i=1}^{m} \left(1 - \mathbb{P}\left(\mathsf{A} \mid p_i, \mathcal{S}_{p_i}\right)\right), \quad (3)$$

$$\mathbb{P}(\mathsf{B}) \approx 1 - \prod_{i=1}^{m} \left(1 - \beta^L\right) = 1 - (1 - \beta^L)^m. \quad (4)$$

Moreover, to numerically quantify the deanonymization probability $\mathbb{P}(\mathsf{B})$ in Nym, we set practical ranges for both the fraction of compromised mixnodes β and the number of transmitted packets m. Regarding β, we note that a mixnode

[3] $M_j^i \xleftarrow{\$} \mathcal{S}_{M_j}$ indicates uniform random sampling of M_j^i from the set $\mathcal{S}_{M_j}$.

adversary may materialize in two ways: (1) as an entity operating multiple mixnodes, possibly distributed across geographic regions, or (2) as a group of mixnodes co-located within the same country or jurisdiction, where legal coercion may compel operators to disclose input–output traffic mappings, ultimately leading to client deanonymization. We focus on the latter scenario in our analysis, as it is empirically measurable.

To this end, using the VerLoc protocol [20] employed in Nym, we derived a dataset containing the geographic locations (latitude and longitude) of Nym mixnodes. The country- and region-level distributions of these nodes are shown in Fig. 1b. As illustrated, the United States currently hosts over 20% of the mixnodes, followed by Finland (19%), Germany (15%), and the Netherlands (6%). At the regional level, more than 50% of nodes are located in Western Europe, 21% in North America, and 9% in Asia. Accordingly, a practical range for the fraction of compromised mixnodes is $0.05 \leq \beta \leq 0.3$.

To set a practical range for m, we note that (1) each Sphinx packet in Nym is 2 KB in size, and (2) for every packet sent from a client to a destination, the destination replies with a return packet routed along a uniformly random path [7]. Therefore, if a client transmits data of size d, the total number of exchanged packets is given by $\frac{2 \cdot d}{2 \text{ KB}}$. Using this formulation, we estimate m by sampling the data volume d over a range from 2 KB to 1 GB.

Under the specified setting and utilizing Eq. (4), we quantify the probability of deanonymization for a mixnet with $L = 3$ layers and $W = 300$ mixnodes, as shown in Fig. 1c, plotted against the amount of data transmitted by clients. As expected, the deanonymization probability increases with the data volume, due to the corresponding rise in the number of transmitted packets, m. Moreover, as the fraction of compromised mixnodes β increases, the probability of deanonymization grows accordingly. Specifically, when the adversary controls 25–30% of the mixnodes, clients are deanonymized after transmitting as little as 256 KB of data. For $\beta = 20\%$ or $\beta = 15\%$—corresponding to node distributions in the US, Germany, or Finland—deanonymization occurs after approximately 1 MB of transmission. When $\beta = 10\%$, the threshold rises to around 4 MB, and for $\beta = 5\%$, which aligns with node concentration in the Netherlands, full deanonymization is observed after roughly 30 MB of transmitted data. These results suggest that clients will inevitably become deanonymized after crossing a particular data volume threshold.

B. Path Selection Schemes

An effective direction for mitigating the advantage of mixnode adversaries is to alter the process of path assignment to packets in $\mathcal{S}_D$.In particular, our goal is to focus this amendment on reducing the dissimilarity among the paths assigned to these packets. This, in turn, decreases the number of distinct mixnodes appearing across the paths, thereby reducing the probability that any given packet traverses a fully compromised path. To this end, we introduce the following strategies in this section: (1) the *K-Hops Fixed* (K-HF) strategy, (2) the *K/W* strategy, and (3) the α-*Sticky Selection* (α-SS) strategy.

1) K-Hops Fixed (K-HF) Strategy: The K-HF strategy reduces dissimilarity among packet paths by fixing a portion of each path assigned to packets in $\mathcal{S}_D$. Specifically, K-HF preselects h_f layers from the mixnet and, from each of these h_f layers, samples one mixnode uniformly at random in advance. Then, the path assigned to each packet in $\mathcal{S}_D$ is constructed by combining the preselected mixnodes with independently and uniformly sampled mixnodes from the remaining (non-preselected) layers. For instance, in Fig. 2a, the first and second hops are fixed to nodes M_1 and M_3, respectively. As a result, all three client packets traverse M_1 and M_3 in the first and second layers, while in the third layer, each packet is independently routed through either M_5 or M_6.

More formally, the K-HF strategy begins by selecting a subset of layers $\mathcal{S}_h \subseteq \mathcal{S}_L$ of size h_f, where $0 \leq h_f \leq L$. For each $k \in \mathcal{S}_h$, a corresponding fixed node h_k is drawn uniformly at random from $\mathcal{S}_{M_k}$. The set of fixed nodes is then defined as $\mathcal{S}_0 = \{h_k \xleftarrow{\$} \mathcal{S}_{M_k} \mid \forall k \in \mathcal{S}_h\}$. Each packet $p_i \in \mathcal{S}_D$ is subsequently assigned a path $\mathcal{S}_{p_i}$, as defined in Eq. (5), which combines the fixed hops in $\mathcal{S}_0$ with mixnodes drawn independently and uniformly at random from the remaining layers $\mathcal{S}_L \setminus \mathcal{S}_h$.

To compute the probability of deanonymization under the K-HF strategy, let A_{h_f} denote the event that all fixed hops on a packet's path are compromised, and let A'_{h_f} denote the event that all non-fixed (randomly selected) hops are compromised. Then, the event A—indicating that a packet's full path is compromised—occurs if both A_{h_f} and A'_{h_f} occur simultaneously. Thus, the resulting deanonymization probability $\mathbb{P}(\mathsf{B})$ is given in Eq. (6). Notably, in Eq. (6), the probability that the non-fixed portion of a path is fully compromised is computed as the complement of the event that, for each packet, at least one mixnode in the non-fixed segment remains honest. By subsequently applying the approximation introduced earlier, Eq. (6) simplifies to Eq. (7).

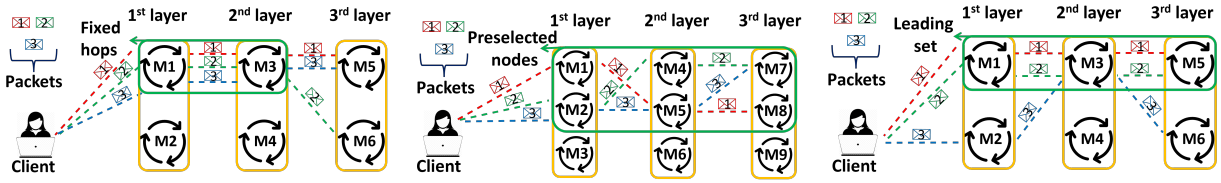
Observe that in Eq. (7), when $h_f = 0$, no hops are fixed, and the expression reduces to Eq. (4). Conversely, when $h_f = L$, all hops are fixed and every packet follows the same path; in this case, the deanonymization probability becomes $\mathbb{P}(\mathsf{B}) = \beta^L$. Further, Eq. (7) yields a lower deanonymization probability than Eq. (4). We formally establish this result in Theorem 1 (Appendix A), demonstrating the effectiveness of the K-HF strategy in mitigating deanonymization risk.

$$\mathcal{S}_{p_i} = \left\{ M_k^i \mid M_k^i \xleftarrow{\$} \mathcal{S}_{M_k},\ \forall k \in \mathcal{S}_L \setminus \mathcal{S}_h \right\} \cup \mathcal{S}_0. \tag{5}$$

$$\mathbb{P}(\mathsf{B}) = \mathbb{P}(\mathsf{A}_{h_f} \mid \mathcal{S}_0) \cdot \left(1 - \prod_{i=1}^{m} \left(1 - \mathbb{P}(\mathsf{A}'_{h_f} \mid p_i, \mathcal{S}_{p_i}) \right) \right), \tag{6}$$

$$\mathbb{P}(\mathsf{B}) \approx \beta^{h_f} \cdot \left(1 - \left(1 - \beta^{L - h_f} \right)^m \right). \tag{7}$$

As discussed earlier, path selection in Nym is designed to maximize the uncertainty faced by the GPA. However, when the diversity of paths among packets in $\mathcal{S}_D$ is reduced, the advantage of the GPA correspondingly increases. To theoret-



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 2: Different path selection schemes.

ically quantify this effect, we consider Shannon entropy [38] as a measure of uncertainty in the probability distribution used for assigning a path to a packet $p_i \in \mathcal{S}_D$, under the assumption that the GPA has full knowledge of the paths $\mathcal{S}_{p_k}$ for all $1 \le k \le i-1$. We denote this conditional entropy by H_D. While H_D provides an upper bound on the GPA's advantage—since the GPA cannot predict previously selected paths with certainty—it serves as a comparative metric.

To evaluate this entropy under the baseline design, let E denote the event that a particular path is assigned to p_i. Since paths are sampled uniformly at random in the baseline, the probability of assigning any specific path to p_i is $\mathbb{P}(\mathsf{E}) = \frac{1}{W^L}$, yielding an entropy of $\mathsf{H}_D = L\log(W)$. Under the K-HF strategy, by contrast, when h_f out of L hops are fixed across all packets, the path support for p_i is reduced to W^{L-h_f}. As a result, the probability of assigning any particular path becomes $\mathbb{P}(\mathsf{E}) = \frac{1}{W^{L-h_f}}$. Assuming that the GPA can infer the fixed hops, the corresponding entropy drops to $\mathsf{H}_D = (L-h_f)\log(W)$. Hence, the GPA's advantage increases by at most $h_f \log(W)$.

2) K/W Strategy: The K-HF strategy reduces the advantage of the mixnode adversary. That said, forcing all packets in $\mathcal{S}_D$ to traverse the same mixnodes at fixed hops may lead to overloading those nodes—particularly during high-volume traffic exchanges between a client and its destination. To mitigate this issue, we introduce the *K/W strategy*. This strategy preselects a subset of K out of the total W mixnodes from each layer. Then, for each packet in the client–destination communication session, a path is assigned by sampling one mixnode uniformly at random from the preselected subset in every layer. For instance, Fig. 2b illustrates a scenario in which the first two out of three available nodes in each layer are preselected. As a result, the client's first, second, and third packets are routed through the paths (M_1, M_5, M_8), (M_2, M_4, M_7), and (M_2, M_5, M_7), respectively.

More formally, the K/W strategy begins by selecting a subset $\mathcal{S}'_{M_j} \subseteq \mathcal{S}_{M_j}$ of size K for each $j \in \mathcal{S}_L$. Then, for each packet $p_i \in \mathcal{S}_D$, a path $\mathcal{S}_{p_i}$ is assigned by sampling one node uniformly at random from the corresponding preselected subset in each layer, as defined in Eq. (8).

Given the preselected sets, we can compute the probability of deanonymization under the K/W strategy using Eq. (9). However, after sampling a subset of nodes in each layer, the fraction of compromised nodes in each subset $\mathcal{S}'_{M_j}$ becomes a random variable, denoted by β_j. The distribution of β_j follows a hypergeometric distribution, determined by W, β, and K. Specifically,

$$\beta_j \sim \text{Hypergeometric}(\beta W, K, W) = \frac{\binom{\beta W}{\beta_j K} \cdot \binom{W-\beta W}{K-\beta_j K}}{\binom{W}{K}}.$$

Let $\boldsymbol{\psi} = \{\beta_j \mid 1 \le j \le L\}$ denote the vector of compromised node fractions across the subsets $\mathcal{S}'_{M_j}$ for $j \in \mathcal{S}_L$. Using this notation, Eq. (9) simplifies to Eq. (10), which represents the deanonymization probability as the expected value over the distribution of $\boldsymbol{\psi}$. While evaluating this expectation exactly requires full knowledge of the mixnet configuration, we can instead upper-bound the expression by noting that the number of possible paths under the K/W strategy is at most K^L. As a result, the deanonymization probability can be derived in Eq. (10) by modifying Eq. (4) accordingly—specifically, by replacing the exponent in $(1-\beta^L)^m$ with $\min(K^L, m)$. This bound demonstrates that the K/W strategy yields a lower deanonymization probability than the baseline, particularly in the regime where $m > K^L$. Furthermore, when $K = W$, Eq. (10) reduces to Eq. (4). Conversely, when $K = 1$, all packets share a fully fixed path, and the deanonymization probability becomes $\mathbb{P}(\mathsf{B}) = \beta^L$.

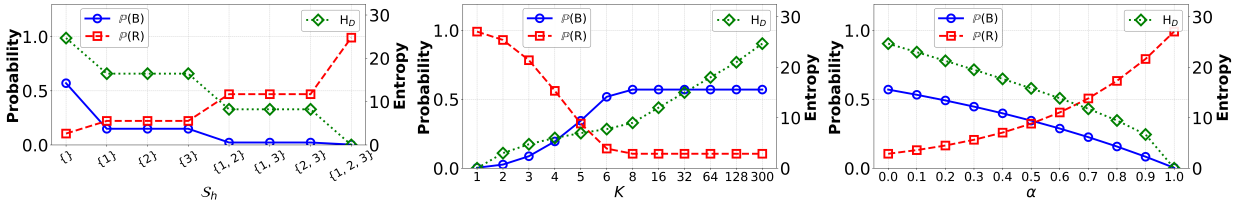
$$\mathcal{S}_{p_i} = \left\{ M_j^i \mid M_j^i \xleftarrow{\$} \mathcal{S}'_{M_j},\ \forall j \in \mathcal{S}_L \right\}, \tag{8}$$

$$\mathbb{P}(\mathsf{B}) = 1 - \prod_{i=1}^{m} \left(1 - \mathbb{P}(\mathsf{A} \mid p_i, \mathcal{S}_{p_i})\right), \tag{9}$$

$$\approx 1 - \mathbb{E}_{\boldsymbol{\psi}} \left[\left(1 - \prod_{j=1}^{L} \beta_j \right)^m \right] \le 1 - (1-\beta^L)^{\min(K^L, m)}. \tag{10}$$

Moreover, under the K/W strategy, if the GPA observes the first $i-1$ packets, the probability of assigning any specific path to packet $p_i \in \mathcal{S}_D$ is $\mathbb{P}(\mathsf{E}) = \frac{1}{K^L}$, yielding a corresponding entropy of $\mathsf{H}_D = L\log(K)$. Compared to the baseline, this results in an increase in the GPA's advantage by $L\log(W/K)$, which can be balanced through appropriate tuning of the parameter K.

3) α-Sticky Selection (α-SS) Strategy: Distinct from the previously introduced strategies that either reduce path diversity by fixing specific hops (K-HF) or limit the selection to only a subset of nodes in each layer (K/W), the α-SS strategy aims to increase similarity among packet paths in $\mathcal{S}_D$ without explicitly excluding any mixnodes from the selection process. Formally, α-SS maintains two sets: the *leading set* $\mathcal{S}_\alpha$ and the *initial set* $\mathcal{S}_I$. At the beginning, the leading set is empty, i.e.,



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 3: Theoretical analysis of different strategies.

$\mathcal{S}_\alpha = \{\}$, while the initial set $\mathcal{S}_I$ includes all possible paths through the mixnet.[4]

Under this setup, the strategy begins by assigning the first packet p_1 a path sampled uniformly at random from $\mathcal{S}_I$, i.e., $\mathcal{S}_{p_1} \xleftarrow{\$} \mathcal{S}_I$. The selected path is then added to the leading set, $\mathcal{S}_\alpha \leftarrow \mathcal{S}_\alpha \cup \{\mathcal{S}_{p_1}\}$, and removed from $\mathcal{S}_I$. For each subsequent packet p_i (where $2 \le i \le m$), a path is selected as follows: with probability α (where $0 \le \alpha \le 1$), a path is sampled uniformly at random from the current leading set $\mathcal{S}_\alpha$ and assigned to $\mathcal{S}_{p_i}$. Conversely, with probability $1-\alpha$, a path is sampled uniformly at random from the initial set $\mathcal{S}_I$ and assigned to $\mathcal{S}_{p_i}$. In the latter case, the selected path is also added to the leading set and removed from the initial set. By construction, at the time of selecting the path for p_i, we have $\mathcal{S}_\alpha = \{\mathcal{S}_{p_1}, \mathcal{S}_{p_2}, \ldots, \mathcal{S}_{p_{i-1}}\}$.

As an example, Fig. 2c illustrates the path assignment process for a communication exchange involving three packets. For the first packet p_1, the assigned path is (M_1, M_3, M_5), sampled uniformly at random from the full path space $\mathcal{S}_I$. This path is then added to $\mathcal{S}_\alpha$ and removed from $\mathcal{S}_I$. When selecting a path for p_2, a Bernoulli trial with parameter α results in reusing the existing path from $\mathcal{S}_\alpha$, so that $\mathcal{S}_{p_2} = \mathcal{S}_{p_1}$. For p_3, if the Bernoulli trial favors selecting a new path, a path—e.g., (M_2, M_3, M_6)—is sampled uniformly at random from $\mathcal{S}_I$ and assigned to p_3.

Furthermore, the deanonymization probability under α-SS depends on the sequential construction of paths, where the path assigned to packet p_i is conditioned on all previously assigned paths $\{\mathcal{S}_{p_1}, \ldots, \mathcal{S}_{p_{i-1}}\}$. Considering this, the formulation for $\mathbb{P}(\mathsf{B})$ is expressed in Eq. (11). Eq. 12 is subsequently derived by simplifying Eq. 11 using Theorem 2, with the complete proof provided in Appendix A. Notably, in Eq. 12, the parameter α governs the degree of path reuse. When $\alpha = 1$, the deanonymization probability reduces to β^L, corresponding to the scenario in which all packets follow the same path. As α decreases, path diversity increases, thereby increasing exposure to compromised mixnodes. In particular, when $\alpha = 0$, the resulting deanonymization probability slightly exceeds that of the baseline (Eq. 4). To ensure that the deanonymization probability under α-SS remains strictly below that of the baseline, Theorem 3 in Appendix A establishes $\alpha \ge \frac{m-1}{W^L+m-2}$.

[4]That is, initially $\mathcal{S}_I = \{M_1 M_2 \cdots M_L \mid \forall j \in \mathcal{S}_L,\ M_j \in \mathcal{S}_{M_j}\}$.

TABLE I: Comparison of different path selection strategies.

Approach	Optimal Parameters	Overall Performance	Complexity	Bandwidth Overhead	Latency
Nym design	NA	0	$\mathcal{O}(1)$	0%	164 ms
K-HF	$h_f = 1$ or $h_f = 2$	[3.1, 3.7]	$\mathcal{O}(1)$	4%	138 ms
K/W	$5 \le K \le 10$	[2.0, 3.4]	$\mathcal{O}(1)$	0.3%	77 ms
α-SS	$0.5 \le \alpha \le 0.9$	[3.4, 5.1]	$\mathcal{O}(m)$	1%	111 ms

$$\mathbb{P}(\mathsf{B}) = 1 - \mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \nsubseteq \mathsf{A}\right) \approx 1 - (1-\beta^L) \tag{11}$$

$$\times \prod_{i=2}^{m} \left(\alpha + (1-\alpha)\left(1 - \frac{W^L}{W^L - (1-\alpha)(i-2) - 1}\beta^L\right)\right). \tag{12}$$

$$\mathsf{H}_D = \alpha \cdot \log\left(1 + (i-2)(1-\alpha)\right) + (1-\alpha) \cdot \log\left(W^L - 1 - (i-2)(1-\alpha)\right). \tag{13}$$

Under the α-SS strategy, the expected number of paths in $\mathcal{S}_\alpha$ before assigning the path for the i-th packet is approximately $1+(1-\alpha)(i-2)$. This estimate arises from the path assignment process: after the initial path is added to $\mathcal{S}_\alpha$, each subsequent packet contributes, on average, $(1-\alpha)$ new paths to the leading set, due to the Bernoulli trial resulting in a new path being selected from $\mathcal{S}_I$ and added to $\mathcal{S}_\alpha$ with probability $1-\alpha$. Consequently, the approximate number of remaining candidate paths in $\mathcal{S}_I$ is $W^L - 1 - (1-\alpha)(i-2)$. Given this setup, the conditional entropy H_D is derived as a weighted sum of the entropy contributions from either $\mathcal{S}_\alpha$ or $\mathcal{S}_I$, quantifying the GPA's advantage, as shown in Eq. (13).

4) Theoretical Comparisons: To compare the strategies theoretically, we focus on their impact on the probability of deanonymization, the advantage gained by the GPA, and the reliability of communication between the client and the destination, particularly in scenarios where each layer contains unreliable mixnodes, any of which—if present along the communication path—renders it unreliable. We denote the probability of reliable communication as $\mathbb{P}(\mathsf{R})$ and refer the reader to Appendix B for its detailed theoretical derivation.

Our comparison, depicted in Fig. 3, is conducted under a mixnet configuration with $L = 3$ layers and $W = 300$ mixnodes per layer, setting $\beta = 0.1$ and $m = 500$ (1 MB of data), with one unreliable node per layer.

Specifically, Fig. 3a presents the results for the K-HF strategy, showing the probability of deanonymization and reliability (measured on the left vertical axis), alongside the entropy H_D (measured on the right vertical axis), across different fixed-hop subsets $\mathcal{S}_h \in \mathcal{S}_L$. As illustrated, when $\mathcal{S}_h = \{\}$ (baseline),

the deanonymization probability $\mathbb{P}(\mathsf{B})$ is relatively high (0.6), reliability is very low (0.1), and the entropy H_D reaches its maximum of 25 bits. As the size of the fixed-hop subset $\mathcal{S}_h$ increases, both $\mathbb{P}(\mathsf{B})$ and H_D decrease due to reduced path diversity, while the overall reliability of the mixnet improves markedly. Notably, fixing just one hop ($h_f = 1$) reduces $\mathbb{P}(\mathsf{B})$ by more than 60%, increases the GPA's advantage by approximately 40%, and doubles reliability relative to the baseline. Further increasing h_f continues to lower $\mathbb{P}(\mathsf{B})$ and improve reliability, although setting $h_f = 3$ yields diminishing returns, as the GPA's advantage approaches its upper bound.

Additionally, the results of the K/W strategy are presented in Fig. 3b for various values of K. As expected, when $K = W$, the behavior closely mirrors that of the baseline. However, decreasing K results in a simultaneous reduction in both the deanonymization probability and the entropy H_D, while reliability increases. Notably, setting $K = 5$ yields a favorable trade-off in this configuration—achieving approximately $4\times$ higher reliability compared to the baseline, a reduced deanonymization probability of $\mathbb{P}(\mathsf{B}) \approx 0.3$, and an entropy value of $\mathsf{H}_D \approx 8$ bits. On the other hand, the performance of the α-SS strategy is illustrated in Fig. 3c. When $\alpha = 0$, the results closely align with those of the baseline. As α increases, both the deanonymization probability and the entropy H_D gradually decrease, while reliability improves due to reduced path diversity. In particular, at $\alpha = 0.7$, the α-SS strategy achieves at least $5.5\times$ higher reliability relative to the baseline, reduces $\mathbb{P}(\mathsf{B})$ to approximately 0.2, and maintains an entropy of $\mathsf{H}_D \approx 11.5$ bits.

To more precisely quantify the optimal settings, we summarize the performance of each strategy in Tab. I, including a comparison with the Nym design (baseline setting). In particular, we identify the best configuration for each method by maximizing a performance score defined as $\mathbb{P}(\mathsf{R}) \cdot (1 - \mathbb{P}(\mathsf{B})) \cdot \mathsf{H}_D$, which favors configurations that achieve low deanonymization probability, high communication reliability, and high entropy H_D. Under this metric, the baseline setting yields a score close to zero, as the reliability is nearly zero and the deanonymization probability is close to one. In contrast, our strategies significantly improve both reliability and anonymity. Notably, the K-HF strategy performs best when $1 \leq h_f \leq 2$, achieving a peak score of 3.7. For the K/W strategy, optimal performance is observed when $5 < K \leq 10$, reaching a maximum score of 3.4. Finally, for the α-SS strategy, selecting α in the range $0.5 \leq \alpha \leq 0.9$ yields favorable performance.

Apart from the performance score, we also compare the proposed strategies based on additional network factors such as client-side path selection overhead, bandwidth overhead for mixnodes, and communication latency. Notably, since Nym is source-routed, clients are responsible for selecting the paths for their communication packets.

Furthermore, because both K-HF and K/W do not require the client to account for previously assigned paths when selecting the path of the ith packet within a session, their path selection overhead per packet remains constant ($\mathcal{O}(1)$), matching that of Nym. In contrast, α-SS requires maintaining and updating the set of previously assigned paths $\mathcal{S}_\alpha$, resulting in higher computational complexity of $\mathcal{O}(m)$ (when $i = m$). In practice, when the communication volume is large, this overhead may become a limiting factor, making K-HF and K/W more efficient alternatives.

On the other hand, path selection in Nym results in each node being selected with equal probability on average. However, our strategic path selection approaches may cause certain nodes to be selected more frequently than others, potentially leading to load imbalances that, in practice, result in communication disruptions due to mixnodes experiencing uneven load and dropping received packets. To evaluate such effects under our approach, we define $\mathcal{S}_U$ as the set of all client–destination pairs. For each mixnode M^* in the mixnet under a given path selection scheme, we measure how frequently it appears along the paths $\mathcal{S}^u_{p_i}$, where $p_i \in \mathcal{S}^u_D$ are the packets belonging to any pair $u \in \mathcal{S}_U$ (as formalized in Eq. (14)). We then compute the average additional load received by nodes whose selection frequency exceeds that of the baseline. As summarized in Tab. I, the K-HF strategy (with $h_f = 2$) results in a moderate bandwidth imbalance of 4%, while α-SS (with $\alpha = 0.8$) incurs a lower imbalance of 1%, and the K/W strategy (with $K = 10$) yields a negligible imbalance of 0.3%.

Lastly, leveraging peer-to-peer latency measurements of nodes in Nym derived via the VerLoc protocol [20], we evaluate the impact of our path selection strategies on communication latency. To this end, let $g(\cdot, \cdot)$ denote the function that returns the link latency between two nodes. Given a packet $p_i \in \mathcal{S}^u_D$ belonging to session $u \in \mathcal{S}_U$, with assigned path $\mathcal{S}^u_{p_i}$, we estimate its end-to-end latency as $\sum_{k=1}^{L-1} g(M^i_k, M^i_{k+1})$, where M^i_k denotes the kth mixnode along the path $\mathcal{S}^u_{p_i}$.

Let $\mathcal{L}_u$ denote the set of per-packet latencies for all packets belonging to session u (see Eq. (15)). The end-to-end latency for session u is then defined as the maximum value in $\mathcal{L}_u$, as the recipient must wait for the arrival of all packets before being able to fully decode the communication message. Based on this quantification, we report the average session-level latency for each strategy in Tab. I, measured for the baseline (Nym), K-HF (with $h_f = 2$), K/W (with $K = 10$), and α-SS (with $\alpha = 0.8$).

As shown, the Nym design yields the highest average latency of 164 ms. In contrast, all proposed strategies result in lower latency, with the K/W strategy reducing latency by more than 50%. This outcome is expected, as the baseline assigns each packet an independent, uniformly random path, which increases the likelihood that at least one packet in $\mathcal{S}^u_D$ will traverse a high-latency route—thereby raising the session's overall latency. In contrast, our path selection strategies increase path similarity among packets within a session, reducing the chance of encountering high-latency paths.

$$\texttt{Load of } M^* = \sum_{u\in\mathcal{S}_U}\sum_{i=1}^{|\mathcal{S}_D^u|}\frac{\mathbb{1}\left(M^*\in\mathcal{S}_{p_i}^u\right)}{\sum_{u\in\mathcal{S}_U}|\mathcal{S}_D^u|}. \tag{14}$$

$$\mathcal{L}_u = \left\{\sum_{k=1}^{L-1} g\left(M_k^i, M_{k+1}^i\right) \;\middle|\; \forall k\in\mathcal{S}_L,\ M_k^i\in\mathcal{S}_{p_i}^u,\ p_i\in\mathcal{S}_D^u\right\}. \tag{15}$$

III. Evaluations

This section presents evaluations based on empirical analysis and simulation, aimed at precisely quantifying anonymity within mixnets. To structure the analysis, we first introduce the evaluation metrics, then describe the experimental setup, and finally present and interpret the results.

A. Anonymity Metrics

Our evaluations in this section are based on both empirical and simulation-based metrics. We first introduce the empirical metrics, which assess the impact of path selection strategies on the advantage gained by mixnode adversaries or the GPA individually, using the DLM and CAM metrics, respectively. We then present the simulation-based metrics, which evaluate the combined effects of mixnode adversaries and the GPA.

Deanonymization Likelihood Metric (DLM). This metric estimates the likelihood that a client–destination pair is deanonymized against a mixnode adversary controlling a subset of nodes in the mixnet. More formally, given a fixed mixnet topology $\mathcal{T}$, a set of adversarial nodes $\mathcal{S}_C$, and a finite set of all client–destination pairs $\mathcal{S}_U$, the DLM assigns a real value $\tau\in[0,1]$ (i.e., $\tau:\mathcal{S}_U\to[0,1]$), estimating the fraction of client–destination sessions that are deanonymized under the given mixnet configuration.

To empirically estimate DLM (i.e., τ), we define the set $\mathcal{S}_{\mathsf{DLM}}^u$ for each client–destination pair $u\in\mathcal{S}_U$ as the set of paths assigned to packets in $\mathcal{S}_D^u$ that are fully composed of adversarial nodes (see formal definition in (16)). A session is considered deanonymized if $|\mathcal{S}_{\mathsf{DLM}}^u|>0$. Thus, the empirical estimate of τ is the fraction of such deanonymized sessions, as shown in (17), averaged over all topologies $\mathcal{T}$, where T in (17) denotes the number of sampled topologies.

$$\mathcal{S}_{\mathsf{DLM}}^u = \left\{\mathcal{S}_{p_i}^u \;\middle|\; \forall j\in\mathcal{S}_L,\ M_j^i\in\mathcal{S}_C\wedge M_j^i\in\mathcal{S}_{p_i}^u\right\}, \tag{16}$$

$$\tau \approx \sum_{\mathcal{T}}\frac{1}{T}\sum_{u\in\mathcal{S}_U}\frac{\mathbb{1}\left(|\mathcal{S}_{\mathsf{DLM}}^u|>0\right)}{|\mathcal{S}_U|}. \tag{17}$$

In other words, DLM, under a mixnode adversary, estimates the likelihood that a sender and receiver become linkable when communicating within a session. In scenarios where all client–destination pairs exchange only one packet, DLM reduces to the fraction of fully compromised paths—a metric originally introduced in [4] and later leveraged in [9], [16], [34], [35]. Additionally, DLM captures the size of the data exchange window during which deanonymization occurs with full certainty (i.e., when $\tau\approx 1$). This is equivalent to the "time-to-first-compromised-stream" metric proposed in [22], although the latter depends on assumptions about user communication patterns that are not easily realizable in practice.

Correlation Advantage Metric (CAM). This metric quantifies the route predictability of packets within client–destination sessions against a GPA that observes all network links. More formally, given a fixed mixnet topology $\mathcal{T}$, a GPA, and a finite set of all client–destination pairs $\mathcal{S}_U$, where each $u\in\mathcal{S}_U$ exchanges a set of packets $\mathcal{S}_D^u$, let $\mathcal{S}_R$ denote the set of all possible paths R_k through the mixnet, where $1\le k\le W^L$.[5] The CAM metric assigns a probability distribution f_R over $\mathcal{S}_R$ (i.e., $f_R:\mathcal{S}_R\to[0,1]$), estimating the probability of each path R_k being selected on average, considering all traffic generated by clients. The Shannon entropy of f_R quantifies the unpredictability of the routes assigned to packets.

To empirically estimate CAM (i.e., f_R), let f_{R_k} denote the probability mass assigned to path R_k, estimated as the fraction of times R_k is selected as $\mathcal{S}_{p_i}^u$, the path assigned to packet p_i in the session of client–destination pair $u\in\mathcal{S}_U$ (see (19) for the formal expression). The entropy of f_R, computed as $-\sum_{R_k\in\mathcal{S}_R} f_{R_k}\log(f_{R_k})$, defines the CAM metric.

$$\mathcal{S}_R = \left\{R_k = M_1M_2\cdots M_L \;\middle|\; \forall j\in\mathcal{S}_L,\ M_j\in\mathcal{S}_{M_j}\right\}, \tag{18}$$

$$f_{R_k} \approx \sum_{\mathcal{T}}\frac{1}{T}\sum_{u\in\mathcal{S}_U}\sum_{i=1}^{|\mathcal{S}_D^u|}\frac{\mathbb{1}\left(\mathcal{S}_{p_i}^u = R_k\right)}{\sum_{u\in\mathcal{S}_U}|\mathcal{S}_D^u|}. \tag{19}$$

In the baseline case, where path selection occurs independently and uniformly at random for each packet, f_R becomes uniform and the entropy (CAM) is maximized—indicating minimal route predictability. Conversely, when CAM approaches zero, the selected paths become highly predictable. However, even in such cases, the GPA can only associate clients with the set of destinations communicating via those predictable paths. Thus, CAM does not lead to sender–receiver linkability but rather captures route predictability. This notion aligns with the definition of route predictability in [34]. CAM, however, extends beyond prior work by quantifying predictability across full communication sessions. In contrast, the "entropy of the transformation matrix" introduced in [34], [35] captures route predictability only in single-packet exchanges, making it a special case of CAM.

Simulation-Based Metric and Packet-Based Entropy. In contrast to the DLM and CAM metrics, simulation-based metrics capture the combined impact of both the GPA and the mixnode adversary. They additionally account for the internal mixing performed within each mixnode, thereby reflecting the effects of both path assignment and intra-node mixing on anonymity. Notably, simulations are implemented using the discrete-event simulator `simpy` [26], written in `Python`. The simulation generates packets on behalf of clients, which traverse mixnodes along their assigned paths, undergo mixing with other packets at each hop, and ultimately reach their intended destinations.

[5] In a mixnet with L layers and W mixnodes per layer, the total number of distinct paths is W^L.

In this setting, one of the state-of-the-art metrics, originally introduced in [24], [3], is *packet-based entropy*. This metric assumes a GPA that observes all network links and additionally has full visibility into the input–output mappings of compromised nodes. It targets a specific packet entering the mixnet and simultaneously monitors all outgoing packets, assigning to each a probability of being the target. Anonymity is then quantified by computing the Shannon entropy of the resulting probability distribution, denoted by $\mathsf{H}(P)$. While this metric, adopted in [16], [34], [35], [31], provides a simulation-based estimate of anonymity, it evaluates packets in isolation. That is, it does not associate packets with specific client–destination pairs, and therefore serves only as an upper-bound measure of anonymity.

Session-Based Anonymity Metric. To address the limitations of packet-based entropy, we introduce a *session-based anonymity* metric that, through simulation and under collusion between the GPA and mixnode adversaries, quantifies session-level anonymity. Formally, given a mixnet configuration $\mathcal{T}$, a set of adversarial nodes $\mathcal{S}_C$, a GPA, a finite set of client–destination pairs $\mathcal{S}_U$, and the set of all packets exiting the mixnet $\mathcal{S}_{\mathsf{Exit}} = \bigcup_{u \in \mathcal{S}_U} \mathcal{S}_D^u$, the session-based anonymity metric assigns to each exit packet $P_E \in \mathcal{S}_{\mathsf{Exit}}$ a probability distribution ν_{P_E} (i.e., $\nu_{P_E} : \mathcal{S}_{\mathsf{Exit}} \to [0,1]^{|\mathcal{S}_{\mathsf{Exit}}|}$), estimating the likelihood that P_E was generated by a particular client–destination session $u \in \mathcal{S}_U$. Let $\nu_{(E,u)}$ denote the probability that packet P_E originated from session u. Then, the session-based anonymity is defined as:

$$\mathsf{H}(S) = -\sum_{\mathcal{T}} \frac{1}{T} \sum_{P_E \in \mathcal{S}_{\mathsf{Exit}}} \frac{1}{|\mathcal{S}_{\mathsf{Exit}}|} \sum_{u \in \mathcal{S}_U} \nu_{(E,u)} \log \nu_{(E,u)}.$$

Session-based anonymity quantifies sender–receiver unlinkability under a joint threat model where the GPA and mixnode adversaries collude. This metric aligns with traditional probabilistic unlinkability models [37], [11], which estimate the effective anonymity set size. Specifically, an entropy value $\mathsf{H}(S) = b$ implies that a given session is, on average, indistinguishable among 2^b possible sessions. Importantly, $\mathsf{H}(S)$ generalizes the packet-based entropy metric $\mathsf{H}(P)$. In scenarios where each client–destination pair in $\mathcal{S}_U$ exchanges only a single packet, $\mathsf{H}(S)$ reduces to $\mathsf{H}(P)$.

To empirically estimate session-based anonymity through simulations, each packet $p_i^u \in \mathcal{S}_D^u$ is assigned, prior to entering the mixnet, a vector ν_i^u of length $|\mathcal{S}_U|$, initialized to all zeros except at index u, where the value is set to 1—indicating the client-destination session to which the packet belongs. In addition, each packet is tagged with a counter $\mathcal{I}_i^u$, initially set to 0, which records the number of compromised mixnodes the packet traverses. This counter is later used to determine whether the packet's corresponding session has been fully deanonymized. Besides this, each mixnode M_j in layer $\ell \in \mathcal{S}_L$ maintains a vector ω_j^ℓ of size $|\mathcal{S}_U|$, initialized to all zeros, which probabilistically tracks the distribution of packets from different client-destination sessions observed within that mixnode.

Considering this initialization, during the simulation process these variables are updated upon two types of events: *receive* (when a packet enters a mixnode) and *send* (when a packet is flushed from a mixnode). Particularly, when a *receive* event occurs and a packet $p_i^u \in \mathcal{S}_D^u$ arrives at a mixnode M_j in layer ℓ, the vector ω_j^ℓ is updated as $\omega_j^\ell \leftarrow \omega_j^\ell + \nu_i^u$, reflecting the arrival of the new packet. If M_j is compromised, the packet's counter is incremented: $\mathcal{I}_i^u \leftarrow \mathcal{I}_i^u + 1$. On the other hand, upon a *send* event, when p_i^u is flushed from M_j in layer ℓ, in case M_j is uncompromised, ν_i^u is updated as $\nu_i^u \leftarrow \frac{\omega_j^\ell}{|\omega_j^\ell|}$, reflecting p_i^u's distribution over possible originating sessions. Additionally, the mixnode vector is updated by subtracting the influence of the departed packet: $\omega_j^\ell \leftarrow \omega_j^\ell - \nu_i^u$.

This process is repeated for all packets across all mixnet layers. Once a packet p_i^u exits the final layer of the mixnet, if $\mathcal{I}_i^u = L$—that is, all mixnodes along the path $\mathcal{S}_{p_i^u}$ were compromised—then the client-destination session to which the packet belongs is considered fully deanonymized. This information is subsequently propagated to exclude the corresponding session from the probability vector ν_i^u associated with other packets. Eventually, for each packet at the mixnet exit, the vector ν_i^u encodes a probability distribution over the possible originating client-destination sessions. The entropy of this distribution, computed for packets in $\mathcal{S}_D^u$, quantifies the anonymity of the corresponding session. Further note that this update mechanism is formally justified in Appendix C, notably through Theorem II, which is derived based on the mixing properties of exponential delay distributions [19]. Additional implementation details are provided in Algorithm 1, along with a toy example, in Appendix C.

Lastly, we note that although the session-level anonymity metric captures the capabilities of a hybrid adversary—combining the strengths of both a GPA and a mixnode adversary—it does not quantify the advantage of a mixnode adversary with only partial network visibility. Our hybrid adversary model, however, provides an upper bound on the advantage that such a partial-view mixnode adversary can achieve.

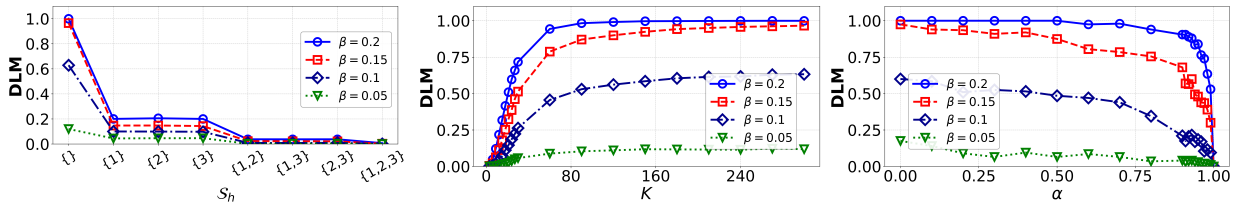
B. Evaluation Setup

Throughout the evaluation, and consistent with Nym's design, we set the number of layers to $L = 3$ and assume $W = 300$ mixnodes per layer, unless otherwise specified. In the simulation, a total of 500 client–destination sessions generate packets according to a Poisson process with a rate of $30{,}000$ packets per second. Each mixnode imposes a delay drawn from an exponential distribution with a mean of 50 ms.[6]

C. Empirical Evaluations

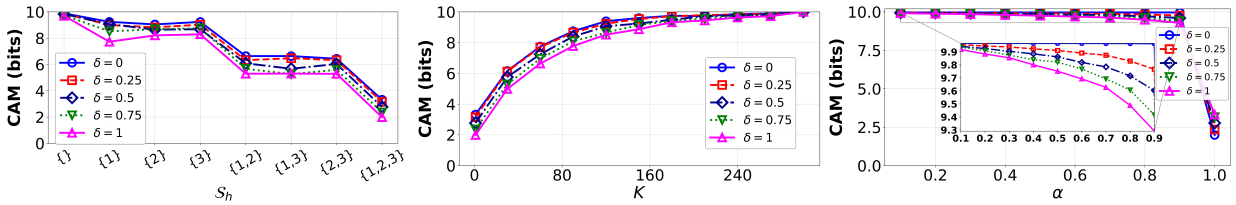
DLM vs. Path Selection Schemes. We begin our analysis by evaluating the DLM metric under different path selection strategies. To this end, we adopt the aforementioned evaluation setup (specifically, $L = 3$ and $W = 300$), and assume that each client-destination pair exchanges approximately 1 MB of data. Additionally, the mixnode corruption fraction β is varied

[6]Our artifact is accessible at https://github.com/whenmixnetsfail.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 4: Evaluation of path selection schemes based on DLM when each client-destination pair exchanges 1 MB of data.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 5: Evaluation of path selection schemes based on CAM.

in the range $0.05 \leq \beta \leq 0.2$, and the corresponding results are presented in Fig. 4.

As evident from Fig. 4, across all strategies, increasing the size of $\mathcal{S}_h$, decreasing K, or increasing α consistently leads to a reduction in DLM. This behavior stems from the fact that each of these adjustments effectively limits the diversity of candidate paths, thereby reducing the likelihood that a packet $p_i^u \in \mathcal{S}_D^u$ traverses a fully compromised route—ultimately resulting in a lower DLM. Conversely, as the corruption fraction β increases, a larger number of mixnodes in each layer become compromised, which increases the total number of fully compromised paths. As a result, even when path similarity is high, the likelihood of assigning a fully compromised path to a packet rises—thereby elevating the DLM across all schemes.

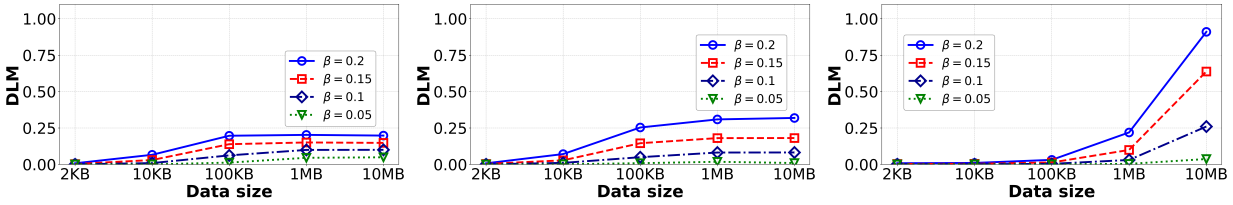
Moreover, Fig. 4a presents the DLM results for the K-HF strategy across different sizes of $\mathcal{S}_h$, showing that fixing even a single hop along the paths assigned to packets in $\mathcal{S}_D$ reduces the DLM by as much as 80% across all values of β. Expanding $\mathcal{S}_h$ to include two fixed hops yields further reductions, and when all hops are fixed (i.e., $\mathcal{S}_h = \mathcal{S}_L$), the DLM approaches zero. For the K/W strategy (Fig. 4b), setting $K \leq 10$—compared to the baseline $K = 300$—results in at least an 80% reduction in DLM. In particular, for $K = 1$, the DLM is nearly zero, while for $10 \leq K \leq 30$, the reduction ranges from approximately 25% to 80%. Finally, Fig. 4c reports the results for the α-SS strategy. When $\alpha \geq 0.9$, we observe substantial reductions in DLM. Specifically, for $\beta \leq 0.1$, the reductions exceed 60%. For higher values of β, the improvements become more gradual but remain significant—especially when $\alpha \geq 0.95$—with the DLM approaching zero when $\alpha = 1$.

CAM vs. Path Selection Schemes. We now investigate how different path selection strategies affect the GPA advantage, as quantified by the CAM metric. To this end, we model the client traffic generation pattern such that the total traffic generated by all clients follows a Poisson process with a global rate μ[7]. Subsequently, each client-destination pair u generates traffic at a rate $q_u\mu$, where q_u denotes the fraction of total traffic attributed to client-destination pair u. We parameterize the traffic distribution by setting $q_u \propto (2^u)^\delta$, for $1 \leq u \leq |\mathcal{S}_U|$ and $0 \leq \delta \leq 1$. When $\delta = 0$, all clients generate traffic at equal rates; that is, each client-destination pair exchanges 1 MB of data. As δ increases, however, higher-index client-destination pairs contribute a disproportionately larger share of the total traffic; i.e., some exchange more than 1 MB, while others exchange less. This skewed traffic model allows us to examine how path selection strategies amplify the GPA advantage under varying traffic distributions. Based on this setup, the CAM metric is evaluated across various strategies, as shown in Fig. 5.

As shown in Fig. 5, increasing the size of $\mathcal{S}_h$, decreasing K, or increasing α consistently reduces CAM across all strategies. This reduction primarily stems from decreased path diversity, which renders the traffic distribution over mixnet paths more predictable and consequently lowers the resulting entropy. Moreover, as the skew parameter δ increases, the CAM metric consistently decreases across all strategies. This effect arises from a higher concentration of traffic among a smaller subset of clients. When combined with path selection strategies, such traffic imbalance causes certain paths to carry disproportionately more traffic than others, thereby reducing the overall entropy of the traffic distribution. Notably, the entropy gap between $\delta = 0$ and $\delta = 1$ reaches up to 2 bits for both the K-HF and K/W strategies, whereas the corresponding reduction under the α-SS strategy remains comparatively modest.

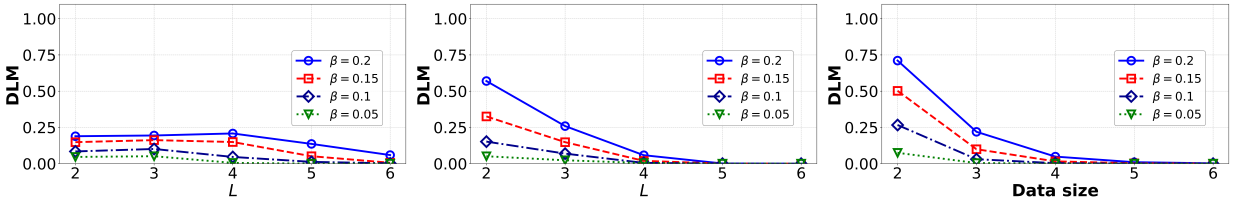
Further analysis in Fig. 5a demonstrates that, under the K-HF strategy, fixing a single hop along each packet's path reduces CAM by up to 2 bits. As the number of fixed

[7]In our evaluation, we set $\mu = 30{,}000$ packets/s.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 6: Data volume vs. DLM: K-HF, K/W, and α-SS are configured respectively with $\mathcal{S}_h = \{1\}$, $K = 10$, and $\alpha = 0.95$.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 7: Impact of the number of layers (L) on the anonymity of the K-HF, K/W, and α-SS strategies, configured respectively with $\mathcal{S}_h = \{1\}$, $K = 10$, and $\alpha = 0.95$, under a data exchange volume of 1 MB.

hops increases to two or three, the reduction becomes more substantial—reaching approximately 5 and 8 bits, respectively. For the K/W strategy (Fig. 5b), setting $K \leq 60$ yields CAM reductions of up to 8 bits relative to the baseline configuration ($K = 300$). However, when $K > 60$, the reduction in CAM becomes less pronounced. Lastly, Fig. 5c presents the results for the α-SS strategy. When $\alpha \geq 0.9$, CAM drops sharply—from 9.3 bits to 2 bits. Outside this high-determinism regime, however, the reduction in CAM remains limited.

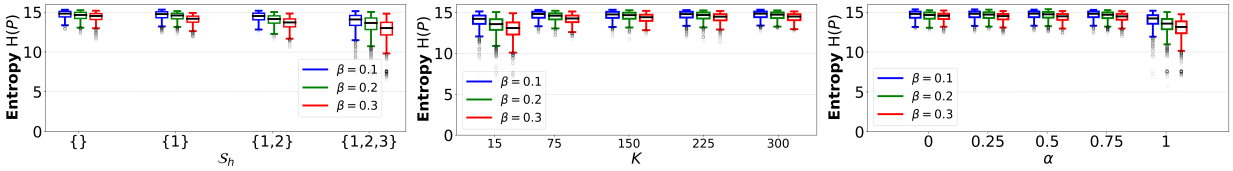
Drawing practical insights from Figs. 4 and 5, we observe that the K-HF strategy achieves up to an 80% reduction in DLM, with a relatively modest CAM cost of approximately 2 bits under the single-hop fixed setting. For the K/W strategy, operating within the effective range $1 \leq K \leq 30$ yields a 50% reduction in DLM, at the cost of a 50% decrease in CAM entropy. The α-SS strategy, by contrast, proves most effective in low-corruption regimes. Specifically, for $\beta \leq 0.1$, it achieves up to an 80% reduction in DLM while incurring only a 1-bit loss in CAM.

Data Size Impact on DLM. Building on the previous analyses, we now examine how increasing the volume of data exchanged between a client and its destination affects the DLM. To this end, we fix $\mathcal{S}_h = \{1\}$, $K = 10$, and $\alpha = 0.95$ under the K-HF, K/W, and α-SS strategies, respectively, and vary the data size from 2 KB to 10 MB. DLM values under these settings are reported in Fig. 6. As shown in Fig. 6, increasing the data size generally results in higher DLM across all strategies. This behavior is expected, as a larger data volume results in a greater number of packets exchanged between each client and destination, thereby increasing the likelihood that at least one packet traverses a fully compromised path. Notably, the increase in DLM is more gradual under K-HF across different values of β. K/W exhibits a slightly steeper increase as data volume grows, while α-SS shows a relatively sharper rise in DLM—likely due to its implicit emphasis on reducing path dissimilarity, which increases the likelihood of packets being routed through fully compromised paths.

These findings reveal that, as data volume increases in practice, both the K-HF and K/W strategies scale more effectively in mitigating deanonymization risk. Moreover, increasing the data volume also amplifies overall network traffic, thereby enhancing the CAM entropy. We present the corresponding CAM results in Appendix D, where we observe that α-SS yields higher CAM entropy compared to both K-HF and K/W.

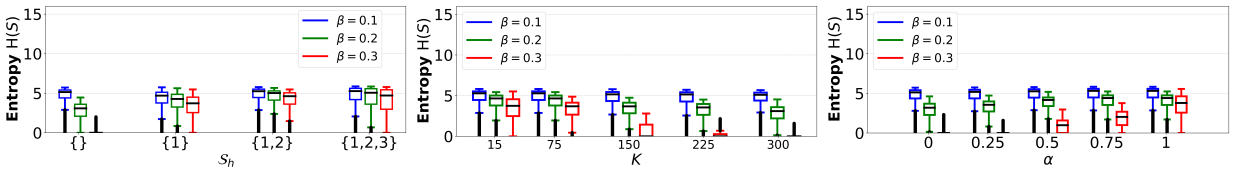
Number of Layers (L) Impact on DLM. Fig. 7 presents a scenario in which we fix $\mathcal{S}_h = \{1\}$, $K = 10$, and $\alpha = 0.95$ under the K-HF, K/W, and α-SS strategies, respectively. Under this configuration, we measure the resulting DLM while varying the number of layers in the mixnet from 2 to 6, with the total amount of data exchanged between each client and destination fixed at 1 MB. The results indicate that increasing the number of layers consistently reduces DLM, as each additional layer expands the space of possible paths, thereby lowering the probability that a given packet is routed through a fully compromised path. Under this configuration, K-HF exhibits a steady and gradual decline in DLM as L increases. In contrast, both K/W and α-SS demonstrate sharper reductions, indicating that these strategies derive greater benefit from deeper mixnets in terms of deanonymization resistance.

Additionally, we provide a corresponding analysis of CAM entropy with varying L in Appendix D. The results suggest that increasing the number of layers leads to higher entropy; thus, increasing L simultaneously reduces DLM and increases CAM. That said, from a practical standpoint, increasing the number of layers results in longer paths, which imposes higher communication latency and increases the risk of unreliable message delivery. Therefore, it is essential to balance the performance costs of deeper mixnets against their benefits.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 8: Entropy $\mathsf{H}(P)$ under different path selection strategies, where each client-destination pair exchanges 1 MB of data.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 9: Entropy $\mathsf{H}(S)$ under different path selection strategies, where each client-destination pair exchanges 1 MB of data.

D. Simulation-Based Evaluations

In this section, we present simulation-based evaluations of our proposed path selection strategies using both the state-of-the-art anonymity metric $\mathsf{H}(P)$ [24], [3] and the session-based anonymity metric $\mathsf{H}(S)$. Simulation results are presented in Figs. 8, 9, 10, and 11, and are visualized using box plots. Each box plot displays the median and the interquartile range (i.e., the 25th to 75th percentiles), with whiskers extending to the 10th and 90th percentiles. Outliers beyond this range are indicated using filled circular markers.

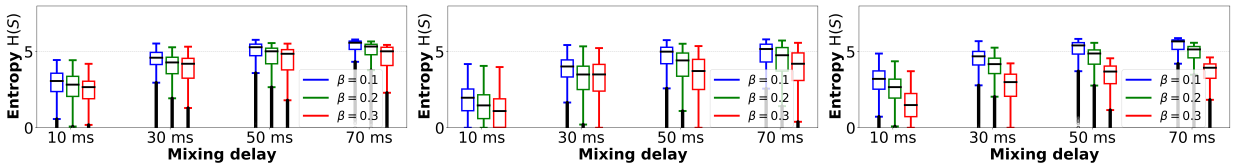
Packet-Based Anonymity. We begin our simulation analysis by quantifying the packet-based anonymity metric $\mathsf{H}(P)$ under our proposed strategies, evaluated across various parameter settings, as shown in Fig. 8, where each client-destination pair exchanges 1 MB of data. As evident from Fig. 8, increasing the corruption fraction β consistently reduces anonymity across all strategies. This is expected, as a higher value of β corresponds to a larger number of adversarial mixnodes, thereby increasing the probability that a given packet traverses one or more compromised hops, ultimately reducing its anonymity.

Beyond this general trend, we observe that all three strategies—K-HF, K/W, and α-SS—yield relatively high packet-level anonymity, with average $\mathsf{H}(P)$ values reaching up to 15 bits. However, when the randomness in path selection is reduced—either by increasing the number of fixed hops in K-HF, decreasing K in K/W, or increasing α in α-SS—the packet-level anonymity exhibits a noticeable decline, in some cases by as much as 3 bits. This behavior aligns with the semantics of $\mathsf{H}(P)$, which quantifies anonymity by measuring the uncertainty in associating each outgoing packet with any of the incoming packets. This uncertainty is maximized when the paths assigned to packets within a session are selected independently and with high randomness. Accordingly, the packet-level results suggest that fully random path selection yields the highest anonymity in mixnets.

Session-Based Anonymity. While packet-based anonymity captures the influence of both the mixnode adversary and the GPA, it evaluates anonymity at the per-packet level, implicitly assuming that each client-destination pair exchanges only a single packet. This assumption is overly restrictive and does not reflect realistic communication patterns, potentially leading to misinterpretation. To address this limitation in greater detail, we present session-based anonymity results under the same simulation setting in Fig. 9. Similar to $\mathsf{H}(P)$, the results for $\mathsf{H}(S)$ indicate that increasing the corruption fraction β consistently reduces session-level anonymity across all evaluated strategies. That said, the entropy values observed under $\mathsf{H}(S)$ are notably lower than those computed using $\mathsf{H}(P)$. Specifically, when $\beta = 0.1$, the average session-based entropy is approximately 5 bits. As β increases to 0.2, the average entropy decreases to the range of 3–5 bits. At $\beta = 0.3$, it further declines, with values ranging approximately from 4 down to 0 bits. This disparity arises because $\mathsf{H}(S)$ quantifies the uncertainty in linking an outgoing packet to its originating client-destination pair, rather than to a specific input packet. Since the number of unique sessions is much smaller than the total number of packets, session-based anonymity naturally yields lower entropy.

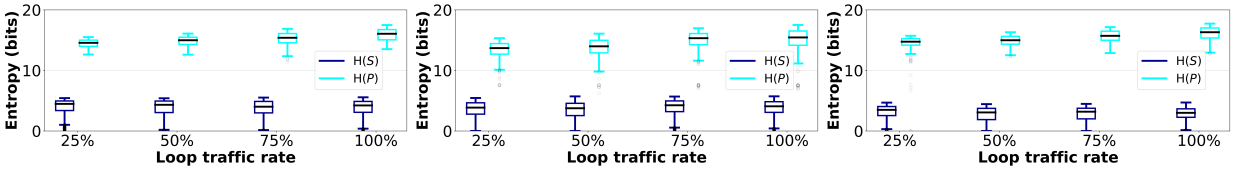
Moreover, increasing randomness among the paths assigned to packets in $\mathcal{S}_D^u$—e.g., by fixing fewer hops in K-HF, increasing K in K/W, or lowering α in α-SS—further reduces $\mathsf{H}(S)$. This trend arises because reduced similarity among paths assigned to packets within a session decreases anonymity. Notably, this behavior stands in stark contrast to the trends observed under $\mathsf{H}(P)$, underscoring that $\mathsf{H}(P)$ serves only as an upper bound and may lead to unsafe design decisions. Finally, under K-HF, fixing even a single hop consistently yields higher session-level anonymity across all values of β compared to the baseline. In contrast, achieving comparable levels of anonymity under K/W and α-SS requires setting $K < 150$ or $\alpha \geq 0.95$, respectively.

Impact of Mixing Process on Anonymity. We additionally analyze the impact of average mixing delay on session-based anonymity—that is, the amount of time each packet is delayed at a mixnode before being forwarded. Specifically, we vary



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 10: Impact of the mixing process on session-level anonymity $\mathsf{H}(S)$ under different path selection strategies.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 11: Impact of clients loop traffic on both $\mathsf{H}(P)$ and $\mathsf{H}(S)$ under different path selection strategies.

the average mixing delay from 10 ms to 70 ms across all path selection strategies. In this experiment, each client-destination pair exchanges 1 MB of data, and we set $\mathcal{S}_h = \{1\}$, $K = 10$, and $\alpha = 0.95$ for the K-HF, K/W, and α-SS strategies, respectively.

The results, shown in Fig. 10, reveal that increasing the average mixing delay consistently improves session-level anonymity across all strategies. This trend arises because longer delays allow packets from different client-destination sessions to be more thoroughly interleaved within the mixnet, thereby enhancing their indistinguishability and increasing overall anonymity. Nonetheless, this anonymity gain comes at a cost, as higher delays introduce greater end-to-end communication latency and increase resource consumption due to prolonged buffering at mixnodes. Hence, selecting an appropriate mixing delay involves a trade-off between anonymity and performance.

Furthermore, we observe that the variance in session-level entropy also increases with longer mixing delays. This effect can be attributed to the long-tailed nature of the exponential delay distribution—which governs the randomization of delays assigned to each packet at mixnodes. As the average mixing delay increases, the distribution of packet exit times becomes wider, ranging from very short to long delays, thereby yielding a broader spread in the resulting entropy values.

Loop Cover Traffic Impact on Anonymity. To conclude this section, we present an additional analysis investigating the impact of loop cover traffic on anonymity—an approach commonly adopted in mixnet designs, wherein clients generate additional packets that are routed back to themselves via the mixnet, with the goal of misleading the GPA [25], [9].

To assess whether this design choice effectively improves anonymity, we evaluate both $\mathsf{H}(P)$ and $\mathsf{H}(S)$ under our proposed path selection strategies. In particular, we assume each client-destination pair exchanges 1 MB of data, and clients additionally inject loop traffic as a fixed fraction of the total transmitted data, ranging from 25% to 100%. For this experiment, we fix $\mathcal{S}_h = \{1\}$, $K = 1$, and $\alpha = 0.95$ for the K-HF, K/W, and α-SS strategies, respectively. The results are shown in Fig. 11.

As Fig. 11 illustrates, increasing the loop traffic rate consistently improves packet-level anonymity, as measured by $\mathsf{H}(P)$, with gains of up to 2 bits as the loop traffic rate increases from 25% to 100%. This may partially explain why such mechanisms are adopted in systems like Nym. However, session-level anonymity—measured via $\mathsf{H}(S)$—remains largely unaffected, with increases of at most 0.3 bits across all settings.

These findings indicate that, in practice, relying solely on $\mathsf{H}(P)$ can be misleading, and highlight the importance of evaluating session-level anonymity via $\mathsf{H}(S)$ to inform the practical deployment of mixnets.

IV. Related Work

In this section, we clarify the novelty of our path selection schemes in relation to prior work on Tor and mixnets.

Relevance to Existing Work on Tor. Mitigating the threat of adversarial nodes through strategic path selection was first proposed for Tor [13]. That said, Tor and mixnets are fundamentally different anonymity systems. Tor assumes a partial network adversary and employs session-based routing, whereas mixnets assume a GPA, adopt per-packet routing, and, unlike Tor, enforce traffic mixing at each hop. Consequently, techniques designed for Tor are not directly transferable to mixnets and, in many cases, are not only inapplicable but can also significantly degrade anonymity in mixnets.

Therefore, while our path selection schemes share some similarities with those proposed for Tor, they are specifically adapted to the requirements and threat models of mixnets. For instance, the K-HF strategy resembles methods in [12], [14], in which the guard relay in Tor remains fixed for extended periods to reduce exposure to adversarial relays. More precisely, in Tor, the guard as well as other relays remain fixed for all packets within a session, and the same guard is reused across multiple sessions. In contrast, mixnets assign a fresh route to each packet within a session. Thus, the K-HF strategy refines the guard concept by fixing one or more hops only

within a session, while ensuring that all hops are resampled for each new session initiated by a client. We emphasize that applying the exact approach used in Tor to mixnets is harmful. Specifically, if all packets in a mixnet session follow the same path, most packets arriving at intermediate mixnodes originate from a small number of sessions. This undermines the mixing effect and substantially increases the GPA's ability to associate packets with their initiators. Furthermore, reusing the same guard across multiple sessions exacerbates this vulnerability.

On the other hand, the K/W strategy limits adversarial exposure by restricting the possible node choices per layer. Weight-based techniques in Tor—such as those based on geolocation or bandwidth [1], [23], [2], [40], [36]—may also be used to restrict node selection per layer. Applying such methods to mixnets, nevertheless, provides a significant advantage to an adversary by increasing the likelihood that their nodes are included in the restricted set—for instance, by injecting artificially high-bandwidth nodes or exploiting geo-based selection biases. K/W avoids this issue by employing randomly preselected nodes per layer. In addition, some of these Tor-based techniques [40], [36] are designed to mitigate partial network adversaries by avoiding paths fully visible to them, making such methods inapplicable to mixnets operating under a GPA.

Lastly, note that in Tor, it has been shown that reusing the same circuit for multiple sessions may improve anonymity [17]. Although this appears similar to α-SS, in mixnets, reusing the same path for all packets within a session (or across sessions) is detrimental to anonymity, as it undermines the mixing process along the path. Thus, α-SS introduces a new approach by probabilistically tuning the path reuse factor to balance resistance against both mixnode adversaries and the GPA.

Relevance to Existing Works in Mixnets. Within the mixnet literature, several path selection schemes have been proposed for low-latency routing [34], [35], [30], [33], [32], [29], [28], [27], which constrain the selection of mixnodes in each layer to those geographically close to nodes in the previous layer, aiming to minimize latency. While this bears some resemblance to the K/W strategy and could, in principle, be used to limit node selection within each layer, their susceptibility to adversarial manipulation—where an attacker can influence the selection of nearby nodes—renders them unsuitable for our purposes.

On the other hand, the only study that acknowledges the practical threat posed by a mixnode adversary is [22]. This work adopts the concept of guard relays from Tor [13], recommending that in a three-layer mixnet, clients fix a node in the second layer as a guard for an extended period across multiple sessions. However, unlike our K-HF strategy, their approach reuses the same guard node across sessions, thereby weakening the mixing process in mixnets and rendering them highly vulnerable to a GPA. Moreover, their method is less generalizable and less flexible compared to K-HF.

V. Conclusion

This work analyzed the deanonymization threat posed by mixnode adversaries in the Loopix mixnet and proposed path selection techniques as mitigations. We introduced novel empirical evaluation metrics and conducted thorough theoretical, empirical, and simulation-based analyses, demonstrating that our techniques effectively mitigate mixnode adversary deanonymization risks without conferring substantial advantages to network adversaries. The lightweight aspects of our approaches make them seamlessly integrable into practical mixnet designs such as Nym.

Acknowledgment

We thank the anonymous reviewers and the shepherd for their insightful feedback and constructive suggestions. This research was partially supported by CyberSecurity Research Flanders under reference number VOEWICS02.

References

[1] M. Akhoondi, C. Yu, and H. V. Madhyastha, "Lastor: A low-latency as-aware tor client," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 476–490.

[2] M. AlSabah, K. Bauer, T. Elahi, and I. Goldberg, "The path less travelled: Overcoming tor's bottlenecks with traffic splitting," in *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings 13*. Springer, 2013, pp. 143–163.

[3] 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.

[4] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, "Denial of service or denial of security?" in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 92–102.

[5] D. Chaum, D. Das, F. Javani, A. Kate, A. Krasnova, J. De Ruiter, and A. T. Sherman, "cmix: Mixing with minimal real-time asymmetric cryptographic operations," in *Applied Cryptography and Network Security: 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings 15*. Springer, 2017, pp. 557–578.

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

[7] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type iii anonymous remailer protocol," in *2003 Symposium on Security and Privacy, 2003.* IEEE, 2003, pp. 2–15.

[8] G. Danezis and I. Goldberg, "Sphinx: A compact and provably secure mix format," in *2009 30th IEEE Symposium on Security and Privacy*. IEEE, 2009, pp. 269–282.

[9] C. Diaz, H. Halpin, and A. Kiayias, "The nym network," 2021.

[10] C. Diaz, S. J. Murdoch, and C. Troncoso, "Impact of network topology on anonymity and overhead in low-latency anonymity networks," in *Privacy Enhancing Technologies: 10th International Symposium, PETS 2010, Berlin, Germany, July 21-23, 2010. Proceedings 10*. Springer, 2010, pp. 184–201.

[11] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Privacy Enhancing Technologies: Second International Workshop, PET 2002 San Francisco, CA, USA, April 14–15, 2002 Revised Papers*. Springer, 2003, pp. 54–68.

[12] R. Dingledine, N. Hopper, G. Kadianakis, and N. Mathewson, "One fast guard for life (or 9 months)," in *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*, 2014, pp. 2–16.

[13] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.

[14] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg, "Changing of the guards: A framework for understanding and improving entry guard selection in tor," in *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, 2012, pp. 43–54.

[15] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 193–206.
[16] I. B. Guirat and C. Diaz, "Mixnet optimization methods," *Proceedings on Privacy Enhancing Technologies*, vol. 1, p. 22, 2022.
[17] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, "Users get routed: Traffic correlation on tor by realistic adversaries," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 337–348.
[18] 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.
[19] D. Kesdogan, J. Egner, and R. Büschkes, "Stop-and-go MIXes: Providing probabilistic anonymity in an open system," in *Proceedings of Information Hiding Workshop (IH 1998)*. Springer-Verlag, LNCS 1525, 1998.
[20] K. Kohls and C. Diaz, "{VerLoc}: Verifiable localization in decentralized systems," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2637–2654.
[21] A. Kwon, D. Lu, and S. Devadas, "Xrd: scalable messaging system with cryptographic privacy," in *Proceedings of the 17th Usenix Conference on Networked Systems Design and Implementation*, 2020, pp. 759–776.
[22] X. Ma, F. Rochet, and T. Elahi, "Stopping silent sneaks: Defending against malicious mixes with topological engineering," in *Proceedings of the 38th Annual Computer Security Applications Conference*, 2022, pp. 132–145.
[23] A. Panchenko, F. Lanze, and T. Engel, "Improving performance and anonymity in the tor network," in *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2012, pp. 1–10.
[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] 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.
[26] Python, "Event discrete, process based simulation for python." https://pypi.org/project/simpy/, 2013.
[27] M. Rahimi, "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
[28] ——, "Larmix++: Latency-aware routing in mix networks with free routes topology," in *International Conference on Cryptology and Network Security*. Springer, 2024, pp. 187–211.
[29] ——, "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.
[30] ——, "Dp-mix: Differentially private routing in mix networks," in *Proceedings of the 41st Annual Computer Security Applications Conference*, 2025.
[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] ——, "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.
[33] ——, "OptiMix: Scalable and distributed approaches for latency optimization in modern mixnets," in *The Network and Distributed System Security Symposium*. Internet Society, 2026.
[34] 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.
[35] ——, "Lamp: Lightweight approaches for latency minimization in mixnets with practical deployment considerations," in *The Network and Distributed System Security Symposium*. Internet Society, 2025.
[36] F. Rochet, R. Wails, A. Johnson, P. Mittal, and O. Pereira, "Claps: Client-location-aware path selection in tor," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 17–34.
[37] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Privacy Enhancing Technologies: Second International Workshop, PET 2002 San Francisco, CA, USA, April 14–15, 2002 Revised Papers 2*. Springer, 2003, pp. 41–53.
[38] C. E. Shannon, "Communication theory of secrecy systems," *The Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.
[39] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes, and C. Diaz, "A survey on routing in anonymous communication protocols," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–39, 2018.
[40] 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.
[41] 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.

APPENDIX

A. Theorems and Proofs

K-HF Strategy. In Section §II, we claimed that if clients within a mixnet employ the K-HF strategy for forming packet paths belonging to $\mathcal{S}_D$, the probability of deanonymization—denoted by $\mathbb{P}(\mathsf{B})$—is reduced compared to the baseline. To formally establish this claim, we present Theorem 1. In the result of this theorem, we substitute $x = h_f$, $\phi(\beta^L) = \mathbb{P}_{\text{Baseline}}(\mathsf{B})$, and $\phi(\beta^{L-h_f}) = \frac{\mathbb{P}_{\text{K-HF}}(\mathsf{B})}{\beta^{h_f}}$, which directly implies that the deanonymization probability under K-HF is strictly lower than that of the baseline.

Theorem 1. *Let $\beta \in (0,1)$ and $x, L, m \in \mathbb{Z}^+$ with $x \leq L$. Then the following inequality holds: $\phi(\beta^L) \geq \beta^x \cdot \phi(\beta^{L-x})$, where $\phi(z) = 1-(1-z)^m$.*

Proof. Define $a := \beta^L$ and $b := \beta^{L-x}$. Since $\beta \in (0,1)$ and $x > 0$, it follows that $0 < a < b < 1$.
Next, define the function $f(z) := \frac{\phi(z)}{z} = \frac{1-(1-z)^m}{z}$, which compares the probability $\phi(z)$ to its input z.
As shown in Lemma 1, $f(z)$ is strictly decreasing over the interval $(0,1)$. Therefore, since $a < b$, we have $f(a) > f(b)$, i.e., $\frac{\phi(a)}{a} > \frac{\phi(b)}{b}$.
Multiplying both sides by a yields: $\phi(a) > a \cdot \frac{\phi(b)}{b}$. Since $\frac{a}{b} = \beta^x$, we conclude that $\phi(\beta^L) > \beta^x \cdot \phi(\beta^{L-x})$, which completes the proof. □

Lemma 1. *Let $f(z) = \frac{1-(1-z)^m}{z}$ for $z \in (0,1)$ and $m \in \mathbb{Z}^+$. Then $f(z)$ is strictly decreasing on $(0,1)$.*

Proof. We apply the quotient rule to differentiate $f(z)$. Let $u(z) = 1-(1-z)^m$ and $v(z) = z$. Then, $f(z) = \frac{u(z)}{v(z)} \Rightarrow f'(z) = \frac{u'(z)v(z)-u(z)v'(z)}{v(z)^2}$. We compute the derivatives: $u'(z) = m(1-z)^{m-1}$ and $v'(z) = 1$. Substituting, we obtain: $f'(z) = \frac{mz(1-z)^{m-1}-(1-(1-z)^m)}{z^2}$. Simplifying the numerator: $f'(z) = \frac{(1-z)^{m-1}(1+(m-1)z)-1}{z^2}$.
Let $g(z) = (1-z)^{m-1}(1+(m-1)z)$. We now show that $g(z) < 1$ for all $z \in (0,1)$, which implies $f'(z) < 0$.
Differentiating $g(z)$: $g'(z) = -m(m-1)z(1-z)^{m-2}$.
Since $m > 0$, $z \in (0,1)$, and $(1-z)^{m-2} > 0$, we conclude $g'(z) < 0$ for all $z \in (0,1)$. Therefore, $g(z)$ is strictly decreasing. Since $g(0) = 1$ and $g(z)$ decreases as z increases, we have $g(z) < 1$ for all $z > 0$.

Thus, the numerator of $f'(z)$ is negative, and the denominator is positive, implying $f'(z) < 0$. □

α-SS Strategy. In this section, we provide the formal expression for the probability of deanonymization under the α-SS strategy. This result is stated in Theorem 2. Furthermore, in Theorem 3, we derive a lower bound on the parameter α that ensures the deanonymization probability under the α-SS strategy remains strictly less than that of the baseline.

Theorem 2. *The probability of deanonymization under the α-SS strategy is given by:*

$$\mathbb{P}(\mathsf{B}) = 1 - \mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \not\subseteq \mathsf{A}\right) = 1 - (1-\beta^L) \times \prod_{i=2}^{m} \left(\alpha + (1-\alpha)\left(1 - \frac{W^L}{W^L - (1-\alpha)(i-2) - 1}\beta^L\right)\right).$$

Proof. To compute the deanonymization probability, we begin by calculating $\mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \not\subseteq \mathsf{A}\right)$, which is the probability that none of the selected paths $\mathcal{S}_{p_i}$ are fully compromised.
Since in the α-SS strategy, the path selected for the i-th packet depends on the $i-1$ previously assigned paths, we apply the chain rule of conditional probability:

$$\mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \not\subseteq \mathsf{A}\right) = \mathbb{P}(\mathcal{S}_{p_1} \not\subseteq \mathsf{A}) \times \mathbb{P}(\mathcal{S}_{p_2} \not\subseteq \mathsf{A} \mid \mathcal{S}_{p_1}), \times \cdots \times \mathbb{P}(\mathcal{S}_{p_m} \not\subseteq \mathsf{A} \mid \mathcal{S}_{p_1}, \ldots, \mathcal{S}_{p_{m-1}}).$$

For the first packet, since the path is sampled uniformly at random from the full path space, the probability that it is not compromised is:

$$\mathbb{P}(\mathcal{S}_{p_1} \not\subseteq \mathsf{A}) = 1 - \beta^L.$$

For the second packet, with probability α, the path is reused from the leading set $\mathcal{S}_\alpha$—specifically, it is equal to $\mathcal{S}_{p_1}$, which is not compromised by construction. With probability $1-\alpha$, the path is sampled uniformly at random from $\mathcal{S}_I$, which now contains $W^L - 1$ paths. Assuming the fraction of fully compromised paths remains approximately β^L, we estimate that around $W^L\beta^L$ out of the original W^L paths are compromised. Consequently, among the remaining $W^L - 1$ paths in $\mathcal{S}_I$, the fraction of compromised paths is approximately $\frac{W^L}{W^L-1}\beta^L$. Thus, the conditional probability that the second packet avoids a fully compromised path, given the first path assignment, is:

$$\mathbb{P}(\mathcal{S}_{p_2} \not\subseteq \mathsf{A} \mid \mathcal{S}_{p_1}) = \alpha + (1-\alpha)\left(1 - \frac{W^L}{W^L - 1}\beta^L\right).$$

Generalizing this to the i-th packet, we observe that approximately $1 + (1-\alpha)(i-2)$ paths have already been removed from $\mathcal{S}_I$ prior to assigning a path to p_i. This includes the path assigned to p_1, and, on average, $(1-\alpha)(i-2)$ additional unique paths that were added to $\mathcal{S}_\alpha$ as a result of Bernoulli trials selecting new paths (with probability $1-\alpha$) in the previous iterations. Consequently, the number of remaining candidate paths in $\mathcal{S}_I$ is approximately $W^L - 1 - (1-\alpha)(i-2)$. Under this setup, the conditional probability that the path assigned to p_i avoids a fully compromised route is:

$$\mathbb{P}(\mathcal{S}_{p_i} \not\subseteq \mathsf{A} \mid \mathcal{S}_{p_1}, \ldots, \mathcal{S}_{p_{i-1}}), = \alpha + (1-\alpha)\left(1 - \frac{W^L}{W^L - (1-\alpha)(i-2) - 1}\beta^L\right).$$

Multiplying over all i completes the expression for $\mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \not\subseteq \mathsf{A}\right)$, and thus the overall deanonymization probability under the α-SS strategy is given by $1 - \mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \not\subseteq \mathsf{A}\right)$, as stated. □

Theorem 3. *The probability of deanonymization under the α-SS strategy is strictly less than that of the baseline when $\alpha \geq \frac{m-1}{W^L+m-2}$.*

Proof. To establish this result, it suffices to determine for which values of α the following inequality holds:

$$\mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \not\subseteq \mathsf{A}\right) \geq (1-\beta^L)^m,$$

which implies that the success probability of avoiding full compromise under α-SS is at least as large as that under the baseline strategy.
From Theorem 2, this probability under α-SS is given by:

$$\mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \not\subseteq \mathsf{A}\right) = (1-\beta^L), \times \prod_{i=2}^{m} \left(\alpha + (1-\alpha)\left(1 - \frac{W^L}{W^L - (1-\alpha)(i-2) - 1}\beta^L\right)\right).$$

The corresponding probability under the baseline strategy is $(1-\beta^L)^m$. Therefore, to ensure the effectiveness of the α-SS strategy, it suffices to show that for all $i \in \{2, \ldots, m\}$,

$$\alpha + (1-\alpha)\left(1 - \frac{W^L}{W^L - (1-\alpha)(i-2) - 1}\beta^L\right) \geq 1 - \beta^L.$$

To determine the threshold α for which this inequality holds with equality, consider:

$$\alpha + (1-\alpha)\left(1 - \frac{W^L}{W^L - (1-\alpha)(i-2) - 1}\beta^L\right) = 1 - \beta^L.$$

Rewriting the inner term:

$$\left(1 - \frac{W^L}{W^L - (1-\alpha)(i-2) - 1}\beta^L\right) = 1 - \frac{1}{A\alpha + B},$$

where we define:

$$A = \frac{i-2}{W^L\beta^L}, \quad B = \frac{W^L + 1 - i}{W^L\beta^L}, \quad D = 1 - \beta^L.$$

These choices satisfy the conditions of Lemma 2, which gives the unique solution for α as:

$$\alpha = \frac{1 - B(1-D)}{A(1-D) + 1} = \frac{i-1}{W^L + i - 2}.$$

This expression represents the minimal α value for which the deanonymization probability at the i-th step under the α-SS

strategy matches that of the baseline. Note that larger values of α provide better protection by increasing the likelihood of path reuse from the uncompromised leading set.
Furthermore, observe that the function $f(x) = \frac{x-1}{W^L+x-2}$ is monotonically increasing in x, which implies that the threshold increases with i. As a result, to ensure that the inequality holds for all $i \in \{2, \ldots, m\}$, it suffices to consider the maximum value, which occurs at $i = m$. Therefore, the sufficient condition for α to guarantee a lower deanonymization probability than the baseline is:

$$\alpha \geq \frac{m-1}{W^L + m - 2}.$$

Under this condition, the probability of deanonymization under the α-SS strategy is strictly smaller than that of the baseline. □

$$\mathbb{P}(\mathsf{R}) = \prod_{i=1}^{m} \mathbb{P}\left(\mathsf{R} \mid p_i, \mathcal{S}_{p_i}\right) = \left[(1-\gamma)^L\right]^m, \tag{20}$$

$$\mathbb{P}(\mathsf{R}) = \mathbb{P}(\mathsf{R}_{h_f}) \cdot \prod_{i=1}^{m} \mathbb{P}(\mathsf{R}'_{h_f} \mid p_i, \mathcal{S}_{p_i}) = (1-\gamma)^{h_f + (L-h_f)m}, \tag{21}$$

$$\mathbb{P}(\mathsf{R}) = \mathbb{E}_{\boldsymbol{\psi}'}\left[\left(1 - \prod_{j=1}^{L} \gamma_j\right)^m\right] \geq (1-\gamma)^{L \cdot \min(K^L, m)}. \tag{22}$$

$$\mathbb{P}(\mathsf{R}) = \mathbb{P}\left(\bigwedge_{i=1}^{m} \mathcal{S}_{p_i} \subseteq \mathsf{R}\right) \approx (1-\gamma)^L \times \prod_{i=2}^{m} \left(\alpha + (1-\alpha)\frac{W^L(1-\gamma)^L - 1 - (1-\alpha)(i-2)}{W^L - 1 - (1-\alpha)(i-2)}\right). \tag{23}$$

Lemma 2. *Let $A, D \in (0,1)$ and $B \in \mathbb{R}_{>0}$. Then the following equation $\alpha + (1-\alpha)\left(1 - \frac{1}{A\alpha+B}\right) = D$ has a unique solution $\alpha \in (0,1)$, given by $\alpha = \frac{1-B(1-D)}{A(1-D)+1}$.*

Proof. We begin by simplifying the left-hand side of the original equation:

$$\alpha + (1-\alpha)\left(1 - \frac{1}{A\alpha + B}\right) = D.$$

Note that: $1 - \frac{1}{A\alpha+B} = \frac{A\alpha+B-1}{A\alpha+B}$, so the equation becomes: $\alpha + (1-\alpha) \cdot \frac{A\alpha+B-1}{A\alpha+B} = D$. Multiplying both sides by $A\alpha + B$, we get:

$$A\alpha^2 + B\alpha + A\alpha + B - 1 - (A\alpha^2 + B\alpha - \alpha) = D(A\alpha + B).$$

Bringing all terms to one side:

$$\alpha\left(A(1-D) + 1\right) + B(1-D) - 1 = 0.$$

To ensure $\alpha \in (0,1)$: note that since $B > 0$ and $D \in (0,1)$, the numerator $1 - B(1-D)$ is positive when $B < \frac{1}{1-D}$. Given this mild constraint on B, the expression is strictly between 0 and 1, ensuring the solution lies in $(0,1)$. □

B. Reliability Assessments

In this section, we examine the impact of our path selection schemes on reliability in the presence of unreliable mixnodes. Such unreliability may arise due to intermittent connectivity, limited computational resources, or adversarial behavior. To abstract this behavior, analogously to the compromise rate per layer, we denote by γ the fraction of unreliable nodes in each layer. Let R denote the event that all nodes along every path assigned to each packet $p_i \in \mathcal{S}_D$ are reliable. Under this setup, for the baseline strategy, the probability that a single path is reliable is approximately $(1-\gamma)^L$. Hence, across m packets, the total reliability, $\mathbb{P}(\mathsf{R})$, is given by Eq. (20). For the K-HF strategy, let R_{h_f} denote the event that the fixed hops are reliable, and let R'_{h_f} denote the event that the remaining hops are also reliable. The resulting reliability expression is given in Eq. (21).

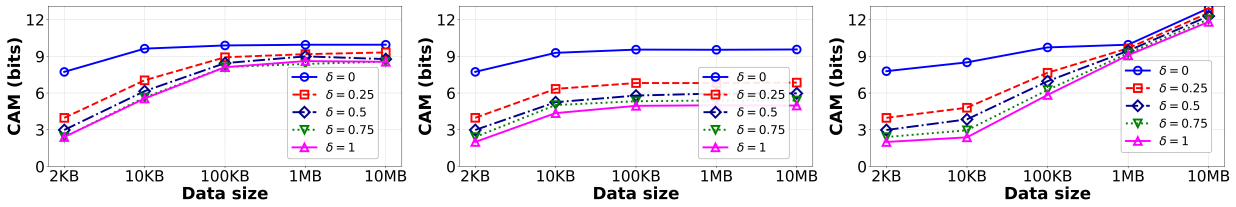
For the K/W strategy, following the analysis in § II (see Eq. 10), let γ_j denote the random variable representing the fraction of unreliable nodes in each pre-selected subset $\mathcal{S}'_{M_j}$ for $j \in \mathcal{S}_L$, and let $\boldsymbol{\psi}'$ denote the vector of all such γ_j values. The total reliability is lower-bounded by Eq. (22). Lastly, for the α-SS strategy, by following an approach analogous to that of Theorem 2, we derive the approximate reliability probability as shown in Eq. (23). Here, $W^L - 1 - (1-\alpha)(i-2)$ denotes the expected number of remaining candidate paths in $\mathcal{S}_I$ after assigning $i-1$ paths, while $W^L(1-\gamma)^L - 1 - (1-\alpha)(i-2)$ gives the approximate number of remaining reliable paths within that subset.

C. Session-Based Anonymity

In this section, we provide further details on the session-based anonymity introduced in § III. Specifically, we focus on updating ν_i^u during the *send* event. In such a case, if M_j is not compromised, the anonymity vector is updated as $\nu_i^u \leftarrow \frac{\omega_j^\ell}{|\omega_j^\ell|}$. This update is theoretically supported by an extension of Theorem II to the setting with $|\mathcal{S}_U|$ client-destination sessions. If M_j is compromised, however, the mixnode adversary has full knowledge of the input-output mappings, and ν_i^u remains unchanged. In both cases, the mixnode's internal vector is updated to remove the influence of the departed packet, according to $\omega_j^\ell \leftarrow \omega_j^\ell - \nu_i^u$.

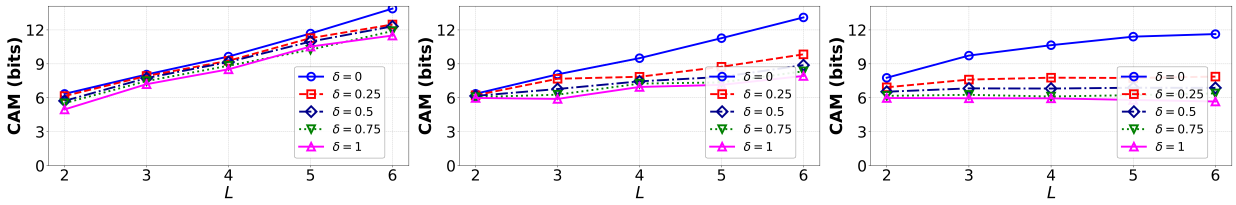
Theorem 4. *Consider two groups of packets existing inside a mixnode, denoted as $\mathcal{S}_{g_1}$ containing N_{g_1} packets and $\mathcal{S}_{g_2}$ containing N_{g_2} packets. If each packet experiences an independent delay drawn from an exponential distribution with parameter λ, then the probability that the next outgoing packet belongs to group one is given by: $\frac{N_{g_1}}{N_{g_1}+N_{g_2}}$.*

Proof. Let T_i denote the delay assigned to the i-th packet, where each $T_i \sim \text{Exp}(\lambda)$ independently. The packet with the smallest delay is flushed first. Let $P^{\mathtt{act}}$ denote the actual outgoing packet, and without loss of generality, assume the first N_{g_1} packets belong to group one. The probability that $P^{\mathtt{act}}$ originates from group one can then be derived as follows:



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 12: CAM vs. m: $\mathcal{S}_h = \{1\}$, $K = 10$, and $\alpha = 0.95$ for the K-HF, K/W, and α-SS strategies, respectively.



(a) K-HF strategy (b) K/W strategy (c) α-SS strategy

Fig. 13: CAM vs. L: $\mathcal{S}_h = \{1\}$, $K = 10$, and $\alpha = 0.95$ for the K-HF, K/W, and α-SS strategies, respectively.

$$\mathbb{P}(P^{\text{act}} \in \mathcal{S}_{g_1}) = \sum_{i=1}^{N_{g_1}} \mathbb{P}(T_i < T_j \ \forall j \neq i), \tag{24}$$

$$= \sum_{i=1}^{N_{g_1}} \int_0^\infty \left(\prod_{\substack{j=1 \\ j\neq i}}^{N_{g_1}+N_{g_2}} \int_{t_i}^\infty \lambda e^{-\lambda t_j}\, dt_j \right) \cdot \lambda e^{-\lambda t_i}\, dt_i, \tag{25}$$

$$= \sum_{i=1}^{N_{g_1}} \int_0^\infty \left[e^{-\lambda(N_{g_1}+N_{g_2}-1)t_i} \right] \lambda e^{-\lambda t_i}\, dt_i, = \frac{N_{g_1}}{N_{g_1}+N_{g_2}}. \square$$

As an example scenario with $|\mathcal{S}_U| = 3$ client-destination sessions and a mixnet consisting of $L = 2$ layers, each containing $W = 2$ mixnodes, consider a packet $p_i^1 \in \mathcal{S}_D^1$ that follows the path $\mathcal{S}_{p_i^1} = \{M_1^1, M_2^1\}$, where only M_2^1 is compromised. Assume the internal vectors of the mixnodes are initialized as $\omega_1^1 = \{0, 2, 1\}$ and $\omega_2^2 = \{1.4, 2.5, 1.65\}$. Under these assumptions, before entering M_1^1, the packet's anonymity vector is $\nu_i^1 = \{1, 0, 0\}$, indicating that it originates from the first communication session. Its compromise counter is initialized as $\mathcal{I}_i^1 = 0$. Upon receiving p_i^u at M_1^1, since M_1^1 is honest, the counter remains unchanged, i.e., $\mathcal{I}_i^1 = 0$, and the mixnode's internal vector is updated as: $\omega_1^1 \leftarrow \omega_1^1 + \nu_i^1 = \{1, 2, 1\}$. Subsequently, during the *send* event, the anonymity vector of p_i^1 is updated as: $\nu_i^1 \leftarrow \frac{\omega_1^1}{|\omega_1^1|} = \{0.25, 0.5, 0.25\}$. Now, p_i^u is associated with session 1 with probability 0.25. The mixnode's internal vector is decremented accordingly: $\omega_1^1 \leftarrow \omega_1^1 - \nu_i^1 = \{0.75, 1.5, 0.75\}$. Next, the packet enters M_2^1, triggering another *receive* event. Since M_2^1 is compromised, the compromise counter is incremented:$\mathcal{I}_i^1 \leftarrow \mathcal{I}_i^1 + 1 = 1$, and the mixnode's vector is updated as: $\omega_2^2 \leftarrow \omega_2^2 + \nu_i^1 = \{1.65, 3.0, 1.9\}$. Upon departure from M_2^1 (a *send* event), the anonymity vector remains unchanged, as M_2^1 is compromised. That is, ν_i^1 remains $\{0.25, 0.5, 0.25\}$. The mixnode's vector is updated accordingly: $\omega_2^2 \leftarrow \omega_2^2 - \nu_i^1 = \{1.4, 2.5, 1.65\}$. Finally, the vector ν_i^1 represents the final distribution over the packet's possible origin across the three sessions. The entropy of this distribution is 1.5 bits, denoted as $\mathsf{H}(S)$.

D. Additional Empirical Analysis

In this section, we provide additional results on evaluating CAM entropy under varying volumes of data exchanged and different numbers of layers. The CAM metric as a function of the amount of data exchanged is presented in Fig. 12, revealing that increasing the data volume exchanged between a client and its destination consistently increases the entropy measured by the CAM metric. This is because a larger number of packets per session tends to uniformize path selection, thereby increasing CAM. Notably, for both the K-HF and K/W strategies, the CAM entropy saturates once the exchanged data volume exceeds approximately 100 KB. This observation suggests that 100 KB may serve as a practical lower bound for mitigating the GPA's inference advantage, while for α-SS, such a threshold is higher. Moreover, Fig. 13 illustrates CAM entropy as a function of the number of layers L, suggesting that increasing the number of layers consistently raises CAM entropy across all strategies. This improvement arises from the fact that deeper mixnets provide greater path diversity and inject additional randomness into packet routing.

E. Artifact Appendix

This artifact appendix provides details on how to reproduce the results presented in the main body of the paper. We leverage two types of evaluation environments: (1) a theoretical setting and (2) an empirical setting.

In the theoretical setting, each proposed technique is evaluated under probabilistic assumptions derived in the methodology sections. In contrast, the empirical setting—implemented in `Python` using the `SimPy` library—involves either numerical estimations of DLM and CAM metrics or full simulations

modeling end-to-end communication flows in mixnets. In the simulation environment, clients generate messages that are routed through a sequence of mixnodes according to the proposed routing strategies. Each mixnode performs mixing operations to anonymize and forward packets, with delays modeled using empirical latency measurements from Nym to capture realistic inter-node transmission characteristics.

The artifact includes approximately 6K lines of `Python` code and reproduces all results presented in the main body of the paper.[8]

1) Description & Requirements: **How to access.** The artifact repository is available via a persistent DOI: https://doi.org/10.5281/zenodo.17703144, and on GitHub: https://github.com/whenmixnetsfail/mix_adversary.

Hardware dependencies. Note that the artifact includes precisely the same configurations and settings presented in the paper. The only difference is that the number of iterations has been scaled down so that the artifact can run on standard systems with 16 GB RAM and 50 GB of disk space (you can also run the artifact on Google Colab).

The estimated runtimes are based on executing the artifact on a system with the following specifications:

- 64-bit CPU: Intel® Core™ i7-10850H @ 2.70 GHz (1 core used)
- Physical Memory (RAM): 16 GB

Software dependencies. Additionally, we tested the artifact on both `Google Colab` and `Ubuntu` 18.04. For users running the artifact on their own servers, we guarantee full functionality on `Ubuntu` 18.04 with `Python` 3.8.10 installed. Prior to running the experiments, it is essential to ensure that all dependencies listed in the `requirements.txt`[9] file are satisfied. We emphasize the importance of adhering to the specified hardware and software dependencies to ensure that the experiments complete within the expected time limits.

Benchmarks. Our evaluation relies on the latency and geographical datasets of the Nym network (https://nym.com), which are included in the artifact repository.

2) Artifact Installation & Configuration: After setting up a system with `Python` version 3.8.10, one can execute the artifact by first installing the required dependencies using the script specified in `requirements.txt`, which is provided in the repository bundled with the code. Upon installing the dependencies, running `main.py` with the arguments explained below will generate the results, which will be saved in the `Figures` directory.

3) Major Claims: The major claims of the paper are summarized as follows:

- (**C1**): The first claim concerns the trend illustrated in Figure 3. Across all settings and scenarios, we observe that as the parameters vary—specifically, increasing the size of $\mathcal{S}_h$ (Figure 3a), decreasing K (Figure 3b), and increasing α (Figure 3c)—the values of H_D and $\mathbb{P}(D)$ on the Y-axis consistently decrease, while $\mathbb{P}(R)$ increases. This claim is substantiated by Experiment E1, which generates Figure 3 and clearly demonstrates this trend.
- (**C2**): The second claim concerns the trend illustrated in Figure 4. Across all settings and scenarios, we observe that as the parameters vary—specifically, increasing the size of $\mathcal{S}_h$ (Figure 4a), decreasing K (Figure 4b), and increasing α (Figure 4c)—the values of DLM on the Y-axis consistently decrease. This claim is supported by Experiment E2, which produces Figure 4 and demonstrates this trend.
- (**C3**): The third claim concerns the trend illustrated in Figure 5. Across all settings and scenarios, we observe that as the parameters vary—specifically, increasing the size of $\mathcal{S}_h$ (Figure 5a), decreasing K (Figure 5b), and increasing α (Figure 5c)—the values of CAM on the Y-axis consistently decrease. This claim is supported by Experiment E3, which produces Figure 5 and demonstrates this trend.

4) Evaluation: This section details the set of experiments conducted to support the main claims presented in the artifact appendix. Executing these experiments produces results stored in the `Figures` directory. Additionally, we explain how to run scripts to regenerate specific results corresponding to figures or tables presented in the main body of the paper.

Experiment (E_1) [Figure 3] ($\leq$ 5 min). This experiment supports claim **C1**.

[Configuration Parameters] The configuration parameters match those used in Fig. 3, specifically: $L = 3$, $W = 300$, $\beta = 0.1$, and $m = 500$ (corresponding to 1 MB of data).

[Preparation and Execution] Run the following command with the `Input` argument set to `100`:

```
python3 main.py
```

[Results] The results will be saved in the `Figures` folder as: Fig. 3a–3c.

Experiment (E_2) [Figure 4] ($\leq$ 40 min). This experiment supports claim **C2**.

[Configuration Parameters] The configuration parameters match those of Fig. 4, specifically: $L = 3$, $W = 300$, and $m = 500$ (1 MB of data).

[Preparation and Execution] Run the following command with the `Input` argument set to `200`:

```
python3 main.py
```

[Results] The results will be saved in the `Figures` folder as: Fig. 4a–4c.

Experiment (E_3) [Figure 5] ($\leq$ 15 min). This experiment supports claim **C3**.

[Configuration Parameters] The configuration parameters match those of Fig. 5, specifically: $L = 3$, $W = 300$, and $\mu = 30{,}000$ packets/s.

[8] At the time of artifact submission, the paper was accepted subject to minor revision; thus, the results provided here are nearly identical to those in the final version.

[9] If you are not using `Python` 3.8.10, you may encounter compatibility errors when running the artifact. In that case, please consider running `requirements_.txt`.

Algorithm 1: Session-Based Anonymity Estimation

Input: Client-destination packet sets $\{\mathcal{S}_D^u\}_{u\in\mathcal{S}_U}$, compromised mixnode set $\mathcal{S}_C$, all mixnodes $\{\mathcal{S}_{M_k}\}_{k\in\mathcal{S}_L}$
Output: Session-based anonymity $\mathsf{H}(S)$
for $u \in \mathcal{S}_U$ **do**
  for $\ell \in \mathcal{S}_L$ **do**
    for *packet* $p_i^u \in \mathcal{S}_D^u$ *entering mixnode* M_j *in layer* ℓ **do**
      Receive Event:
      if $M_j \in \mathcal{S}_C$ **then**
        $\mathcal{I}_i^u \leftarrow \mathcal{I}_i^u + 1$
      end
      $\omega_j^\ell \leftarrow \omega_j^\ell + \nu_i^u$
      Send Event:
      if $M_j \notin \mathcal{S}_C$ **then**
        $\nu_i^u \leftarrow \frac{\omega_j^\ell}{|\omega_j^\ell|}$
      end
      $\omega_j^\ell \leftarrow \omega_j^\ell - \nu_i^u$
    end
  end
end
for *each packet* $p_i^u \in \bigcup_{u\in\mathcal{S}_U}$ *exiting the final layer* **do**
  Compute entropy: $\mathsf{H}_i = \text{Entropy}(\nu_i^u)$
end
Compute overall session anonymity: $\mathsf{H}(S) = \sum_{p_i^u\in\mathcal{S}_D} \mathsf{H}_i$

[Preparation and Execution] Run the following command with the `Input` argument set to `300`:

```
python3 main.py
```

[Results] The results will be saved in the `Figures` folder as: Fig. 5a–5c.

Experiment (E*) [All Others] ($\leq$ 1 h). This experiment allows the user to generate any figure or table shown in the main body of the paper, regardless of whether it is linked to a claim.

[Preparation and Execution] Refer to Table II for the corresponding `Input` argument value and run:

```
python3 main.py
```

[Results] Figures will be saved in the `Figures` folder. Table data should be printed directly in the terminal.

5) Parameter Settings and Execution Time: The network parameter settings are consistent across all experiments and reflect the configurations described in the evaluation section of the paper. These parameters are initialized in `Main_Functions.py` and do not require manual changes to reproduce the core results.

However, for users interested in exploring additional experiments or tuning performance, a subset of parameters can be safely adjusted. Note that mixnets are highly interdependent systems, and improper configuration may lead to invalid results or unexpected runtime behavior. In particular, some values are derived from prior studies to ensure consistency and fairness across experimental settings.

TABLE II: Mapping of input arguments to specific figures and tables.

Experiment	Input	Experiment	Input
Fig. 1	1	Fig. 3	3
Fig. 4	4	Fig. 5	5
Fig. 6	6	Fig. 7	7
Fig. 8a & 9a	891	Fig. 8b & 9b	892
Fig. 8c & 9c	893	Fig. 10a	101
Fig. 10b	102	Fig. 10c	103
Fig. 11a	111	Fig. 11b	112
Fig. 11c	113	Tab. 1	1000

Safely Adjustable Parameters in Main_Functions.py:

- `Iterations` $\rightarrow$ Controls the number of simulation iterations. This can be increased (e.g., up to 30) to improve statistical accuracy; however, note that the computational cost grows approximately linearly.
- `num_targets` $\rightarrow$ Specifies the number of target messages in the simulation. Acceptable values range from 20 to 200.
- `run` $\rightarrow$ Sets the duration of each simulation time slot. Can be tuned between 0.3 and 1.0 (real values).
- `delay1` $\rightarrow$ Represents the average delay introduced at each mixnode. This can be modified within the interval $[0.01, 0.08]$ to simulate different latency conditions.

Other parameters in `Main_Functions.py` should **not** be modified unless the user has deep familiarity with the internal logic and dependencies of mixnet protocols. Inappropriate changes may compromise result validity or break simulation behavior. Users seeking to explore such changes are encouraged to contact the authors for further guidance.