Blending Different Latency Traffic With Beta Mixing

Iness Ben Guirat*
COSIC, KU Leuven
Belgium
ibenguir@esat.kuleuven.be

Debajyoti Das*
COSIC, KU Leuven
Belgium
debajyoti.das@esat.kuleuven.be

Claudia Diaz
COSIC, KU Leuven
Nym Technologies SA
Belgium
cdiaz@esat.kuleuven.be

ABSTRACT

We analyze the anonymity provided by continuous mixnets (e.g., Loopix) when messages with different latency requirements are sent through the same network. The anonymity provided by existing mixnets that offer bounded latency guarantees has only been studied considering that all the traffic in the network follows the same latency distribution. In this work we evaluate whether it is beneficial to aggregate different types of traffic in the same network compared to keeping them separate, when the latency distributions are exponential and the traffic arrivals are a Poisson process — as is the case in Loopix and related designs. We present a novel evaluation method to analyze the leakage to the adversary when multiple different types of traffic are sent through the same network of continuous mixes. We apply the method to empirically evaluate the end-to-end anonymity (in terms of entropy) for each type of traffic in the presence of a global passive adversary that may additionally compromise a constant fraction of mixes or may have knowledge about the type of traffic of network output messages. Finally we show via empirical evaluation using our analytical framework that it is beneficial for anonymity to blend different types of traffic in the same mixnet.

KEYWORDS

anonymous communication, mixnet

1 INTRODUCTION

Over the past few decades, a wide range of literature has emerged discussing Anonymous Communications Networks (ACNs) [1, 6, 7, 10, 20–22, 24], paralleled by the deployment of real-world mixnet-based systems [12]. Mix networks, or mixnets, are a variant of Anonymous Communications Networks (ACN) [1, 6, 7, 10, 20–22, 24] that were designed to protect against traffic correlation by global adversaries who can observe all the traffic in the network. They reroute the traffic through multiple servers known as *mixnodes*, that delay and reorder the messages before forwarding them, in order to hide the correlation between the input and output messages of the mixnet. While there exist a variety of mixing strategies in the literature [14], such as threshold and timed mixing, they either add an unpredictable end-to-end latency or they provide anonymity that do not take traffic levels into consideration. For instance, threshold mixing waits for a threshold number of messages before the messages are forwarded, and therefore, adds an unpredictable end-to-end latency. Designs based on timed mixing flush out messages at predetermined time intervals, even if there are only a few messages inside the mixnodes. Tuning these parameters, such as a smaller threshold or higher time intervals, according to the needs of different applications is a tradeoff between latency and anonymity as shown in [8, 9]. On the other hand, continuous mixnets [12, 22] based on a stop-and-go mixing strategy adds a random delay (typically from exponential distribution) on each hop of a message, independent of other messages, to offer a predictable end-to-end latency as well as an average anonymity that is correlated with the traffic amount.

Therefore, this stop-and-go mixing strategy allows blending different types of traffic, each having a different latency requirement in the same mixnet, simply by drawing the different delays for each traffic type from different distributions. In this work, we investigate the implications on anonymity of blending different traffic types, each having a different latency requirement, in a single mixnet. We call the strategy of blending different traffic with different latency requirements *beta-mixing*.

The impact of blending different types of traffic on anonymity guarantees of the mixnet is an open question — all the existing analyses on continuous mixnets [3, 5, 11, 15, 17, 22] consider that all the messages delayed according to the same distribution. Their techniques are not adequate to handle the scenario when the delays for different messages are drawn from different distributions. This work aims to address this problem and answer relevant questions related to continuous mixes when multiple traffic types with different latency requirements are blended together:

(1) How do we quantify the anonymity provided by the mixnet when different types of traffic are blended together?
(2) Are there any advantages or disadvantages to anonymity when different types of traffic with different latency requirements are blended through the same mixnet, compared to routing each type via a separate mixnet?
(3) Do other factors such as the number of layers, the number of mixes per layer, the rate delays and the traffic generations rates of each traffic type have an impact on each or both traffic types when blended together?

1.1 Contributions

This work provides the first quantitative analysis on anonymity when multiple different types of traffic are blended through a mixnet. Our results demonstrate that blending multiple traffic types does not harm the anonymity guarantees, rather improves for continuous mixnets when the latency distributions on the mixnodes

*These authors contributed equally to this work.

Proceedings on Privacy Enhancing Technologies 2024(2), 464–478

https://doi.org/10.56553/popets-2024-0059

are exponential and the traffic arrivals are poisson processes — as is the case in Loopix [22] and related designs [12].

As part of our analysis technique, we present novel mathematical foundations to analyze the correlation between input and output messages of an honest mixnode in the presence an adversary that observes all the traffic in the network and derives probabilistic relationships between network inputs and outputs. Even though messages from multiple different types of traffic could potentially mix in an honest mixnode, an adversary might still be able to separate the traffic based on their individual delays, which might help the adversary track a specific message. We show that there is indeed a leakage to such an adversary, and quantify the leakage as a function of the distribution parameters of the traffic types and the observed delays (§3).

Based on our proposed mathematical foundations, we evaluate the anonymity in terms of entropy for the entire mixnet using empirical evaluations (§4), with an illustrative example of two different types of traffic. Our evaluation results provide important insights demonstrating the benefits of blending two different types of traffic over sending them through separate mixnets:

(1) Blending improves anonymity when the adversary can see the type of all messages entering the mixnet, but not the types of the messages exiting the mixnet.
(2) If the adversary can observe the types of all incoming and outgoing messages, blending does not offer any advantages nor disadvantages.
(3) The delay parameter for one traffic type impacts not only the anonymity of messages within that specific type but also of those from the other type.
(4) Even when a single message from one traffic type is blended with messages all belonging to the other type, that message achieves significant anonymity (compared to zero anonymity when it is not blended).

The insights from the results with two types of traffic are immediately translatable to the case with more than two types (§4.6). Our methods and results allow protocol designers to quantitatively evaluate the anonymity guarantees of supporting multiple applications with different latency requirements through the same mixnet deployment, and shows that it is beneficial to do so. Especially for an application with very few users, blending with other traffic provides tremendous advantage compared to not blending (which would provide almost no anonymity because of the scarcity of messages).

1.2 Related Works

In an interesting work by Dingledine et al. [16], the authors propose a technique called “alpha-mixing” to mix messages with different latencies in a mixnet. Their technique proposes a hybrid mix batching strategy to integrate users with diverse anonymity and performance goals. Senders allocate a security parameter "α" to each mix in a message’s route, determining its time in each mix. Users can enhance their anonymity by increasing α, and the overall anonymity of the network increases. However, their technique is restricted to traditional deterministic batching strategies and cannot provide predictable latency for the messages. Moreover, they do not provide any quantitative anonymity analysis.

Continuous mixnets [12, 22] are particularly suited for meeting latency constraints, since the sender encodes the delays for each mixnode and can thus predict the overall latency of a message. They can, therefore, support multiple traffic with different delay distributions. In fact it is already possible to use mixnet-based network Nym [12] for Telegram and crypto-currency transactions. To the best of our knowledge, all existing analyses [3, 11, 17, 22] of continuous mixnets focus on a single type of traffic following the same exponential delay distribution. Therefore these anonymity evaluation techniques are not suitable for situations involving multiple traffic types. Even though continuous mixnets are around for two decades, our work closes the above gap for the first time by providing the mathematical tools and quantitative evaluations.

2 PROBLEM STATEMENT AND OVERVIEW

2.1 System Model

Users. We consider a user population $\mathcal{U}$ where each user generates traffic independently of other users. Message generation for each traffic type i follows a Poisson distribution with rate λ_i' messages per user second. We note that only messages generated by honest users (not controlled by the adversary) are relevant to anonymity, and thus $\mathcal{U}$ excludes malicious users. The overall network traffic generation is therefore $\mathcal{U}$. $\sum_{i=1}^{n_\mathcal{T}} \lambda_i$, where $n_\mathcal{T}$ is the number of traffic types blended in the same mixnet.

Source routing. We consider a source-routed continuous mixnet [12, 22], where the sending user chooses the sequence of overlay mixnodes that compose the route of a message, as well as the mixing delay to be applied to the message by each intermediary mixnode on its route.

Mixing Delays. A message’s per-mix delays are drawn as independent samples from an exponential distribution [18]. The mean of the used exponential distribution depends on the type of application and its latency tolerance. The per-mix delays are encoded by the sender in the message headers. Upon decrypting a received message, a mix node retains the message in its internal memory for the specified delay, before proceeding to forward it to its next destination in the route.

Topology. We consider a network topology where mixes are arranged in L ordered layers. The layers are interconnected such that each mix in layer i receives messages from mixes in layer $i-1$ and sends messages to mixes in layer $i+1$; while the first layer receives messages from senders and the last layer forwards messages to their final recipients. The path length of message routes is determined by the number of layers L. To select a message route, each user chooses the nodes for each message uniformly at random from each layer. Prior work has found that layered network topology provides better anonymity properties than free routes [13].

2.2 Beta-mixing

We consider the scenarios where users are using different applications and send their traffic via the mixnet. This is already the case in the Nym network [12], where users are able to send their Telegram traffic as well as cryptocurrency transactions using the same network. Currently the Nym network is using the same default delay parameters for both of the traffic types. However, users have higher latency tolerance for cryptocurrency transactions reaching

10 minutes for bitcoin [1], and lower latency tolerance for instant messages. We denote each traffic type by $\mathcal{T}_i$), the delays are chosen from exponential distribution with rate parameter λ_i. Overall, the total amount of $\mathcal{T}_i$ traffic entering the network follows Poisson distribution with parameter λ'_i messages per second. We summarize the notations in Table 1.

$\mathcal{T}_i$	i-th traffic type
λ'_i	rate generation of traffic type $\mathcal{T}_i$
λ_i	parameter of the exponential distribution for $\mathcal{T}_i$
L	number of layers in the mixnet
W	width (number of mixnodes per layer) of the mixnet
k	total number of messages in a given mixnode
k_i	number of messages of type $\mathcal{T}_i$ in the mixnode
$\mathbf{E}(X)$	expectation of a random variable X

Table 1: Notations

2.3 Attacker Model And Security Goals

The adversary observes all the traffic exchanged in the network links. We assume the adversary monitors the network from the first message sent. Additionally, the adversary may compromise a fraction of the nodes (e.g., 10% of all nodes are compromised, and 90% nodes are honest). The compromised nodes are *honest but curious* — i.e., they still route messages following the protocol specifications but leak to the adversary their internal state, which per message includes the amount of delay applied in the compromised node, and its immediate predecessor and successor in the message route. Finally, we also consider an adversary who knows the type of traffic of all the network output messages.

Compromised users and active attacks. For our analysis we assume that all the senders are honest. Note that compromised senders that leak to the adversary information about their messages simply achieve that their fully traceable messages do not contribute to the anonymity of honest senders, but still cannot undermine the anonymity that honest users provide to each other. Thus, the anonymity of messages from honest users in a scenario with compromised users is simply equivalent to only considering the messages sent by the subset of honest users. We do not consider any active attacks, noting that the relevant active attacks and corresponding defense strategies mentioned in Loopix [22] are applicable, independently of using a single or multiple mixing delay distributions. We thus consider active attacks to be orthogonal to the analysis presented in this work.

2.4 Anonymity Metric

We evaluate anonymity using the entropy metric [15, 23]. Although indistinguishability based metrics [2, 19] are suitable for measuring worst-case scenarios, entropy-based metrics are better suited to capture the effect of network scaling (in terms of anonymity set size) on average anonymity. An entropy of, e.g., 10 bits, indicates that a message is as anonymous as if it was perfectly indistinguishable among about a thousand ($2^{10} = 1024$) other messages, while 11 bits correspond to perfect indistinguishability among $2^{11} = 2048$ messages. Note that the scale is logarithmic, and that an increase of one bit of entropy doubles the size of the equivalent perfect indistinguishability set, while a drop of one bit halves it.

[1] https://medium.com/klaytn/a-comparison-of-blockchain-network-latencies-7508509b8460

2.5 Overview of Evaluation Strategy

In order to evaluate the entropy for an input message to the mixnet, we need to derive the probabilities that correlates the input message to the output messages. We do that in two steps: (1) first, we derive the correlation between the input and output messages of an honest mixnode based on the observations of the adversary; (2) then, based on the above mathematical derivations, we experimentally evaluate the probabilities correlating the input and output messages of a large mixnet.

When there is a single type of traffic, all the messages inside a node are equally likely to be the next message to come out next [3, 11, 17, 22]. However, when many types of traffic are blended, the adversary might be able to partially guess the type of an outgoing message (fast traffic tend to go out earlier than slow traffic); and that would allow the adversary to correlate messages given some background knowledge about the types of the input messages. In the next section, we derive the probabilities that correlate one input message to output messages of one honest mixnode. Finally, in Section 4, we employ this probability distribution within the entropy metric to evaluate various configurations of mixnets.

3 ANALYSIS FOR A SINGLE MIXNODE

In this section we derive the probabilities connecting the input and output messages of a single standalone honest mixnode. For the ease of explanation, we derive the probabilities in the following steps: (1) first consider the most simple case when the adversary knows the types of all messages inside the mixnode; (2) then we derive the probability distributions for the number of messages of each type (assuming only two types) when the adversary knows the types of all incoming messages, but does not know which of them are still in the mixnode; (3) in Section 3.3, we extend our analysis for a more general case (still with two types) where the adversary knows the types of incoming messages only with certain probabilities; (4) finally in Section 3.5, we provide the full derivation for more than two types.

While the traffic type entering the mixnet is not immediately visible, it may be possible to infer it for the first layer, for example if the message sending rate is indicative of which application may be generating the traffic. For the second and subsequent layers, that information is not available to the adversary; however, as we will see in our derivations shortly, the adversary might be able to guess the types of messages with certain probabilities based on the observed delays in the previous layer.

3.1 Very Simple Case

As mentioned above, we first consider the most simple case where the adversary knows the types of all messages inside the mixnode. We assume that the outgoing traffic type is not directly available to the adversary, and thus we want to calculate the probability that a specific outgoing message is a target input message of a known type when the adversary does not know the types of the outgoing

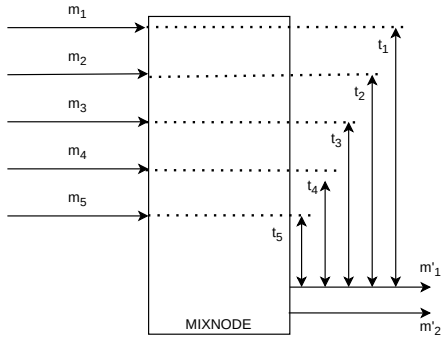


Figure 1: Adversarial observation around an honest mixnode where the mixnode receives messages m_1, m_2, m_3, m_4, m_5 at different times, and a message m'_1 goes out of the mixnode with relative time differences t_1, t_2, t_3, t_4, t_5 respectively.

messages. Since the delays of the different messages are chosen from different delay distributions, based on the observed delays the adversary can partially guess which outgoing message may be of which type.

Consider the example in Figure 1: five messages m_1, m_2, m_3, m_4, m_5 are received by the mixnode at different times, and the adversary knows the types of each of them. No other messages have been received by the mixnode and no messages have left the mixnode yet. Then the first output message m'_1 goes out of the mixnode with relative time differences t_1, t_2, t_3, t_4, t_5 with each of the input arrivals, respectively. Suppose one of those five incoming messages, e.g., m_1 was sent by the target user, and the adversary wants to track that message. Let us assume that $m_1 \in \mathcal{T}_1$. We denote $m_1 = m_{\texttt{target}}$. We want to calculate the probability that $m'_1 = m_{\texttt{target}}$.

Given that m_1 did not go out before t_1, the probability that m_1 has a delay in $[t_1, t_1 + \Delta t)$ (assuming $m_1 \in \mathcal{T}_1$) can be written as follows:

$$\begin{aligned}
&\Pr\left[\text{delay}(m_1) < t_1 + \Delta t \mid \text{delay}(m_1) >= t_1\right] \\
=&\frac{\Pr\left[\text{delay}(m_1) < t_1 + \Delta t \ \wedge \text{delay}(m_1) >= t_1\right]}{\Pr\left[\text{delay}(m_1) >= t_1\right]} \\
=&\frac{\int_{t_1}^{t_1+\Delta t} \lambda_1 e^{-\lambda_1 x} dx}{\int_{t_1}^{\infty} \lambda_1 e^{-\lambda_1 x} dx} \\
=&\frac{e^{-\lambda_1 t_1} - e^{-\lambda_1 (t_1+\Delta t)}}{e^{-\lambda_1 t_1}}
\end{aligned} \tag{1}$$

To generalize, let us assume that there are a total of d types of traffic and total k_j input messages of each type $\mathcal{T}_j$, $j \in \{1, 2, \ldots, d\}$. After the observations until time t_1 (denoted as O) of the incoming messages and the first outgoing message, the probability that m'_1 is a specific message m_1 can be calculated as follows:

$$\begin{aligned}
&\Pr[m'_1 = m_1 | m_1 \in \mathcal{T}_1 \wedge O] \\
=&\frac{\Pr\left[\text{delay}(m_1) = t_1 | m_1 \in \mathcal{T}_1 \wedge \text{delay}(m_1) >= t_1\right]}{\sum_{j=1}^{d} \sum_{m_i \in \mathcal{T}_j} \Pr\left[\text{delay}(m_i) = t_i | m_i \in \mathcal{T}_j \wedge \text{delay}(m_i) >= t_i\right]} \\
=&\lim_{\Delta t \to 0} \frac{\Pr\left[\text{delay}(m_1) < t_1 + \Delta t | m_1 \in \mathcal{T}_1 \wedge \text{delay}(m_1) >= t_1\right]}{\sum_{j=1}^{d} \sum_{m_i \in \mathcal{T}_j} \Pr\left[\text{delay}(m_i) < t_i + \Delta t | m_i \in \mathcal{T}_j \wedge \text{delay}(m_i) >= t_i\right]} \\
=&\lim_{\Delta t \to 0} \frac{\frac{e^{-\lambda_1 t_1} - e^{-\lambda_1 (t_1+\Delta t)}}{e^{-\lambda_1 t_1}}}{\sum_{j=1}^{d} \sum_{m_i \in \mathcal{T}_j} \frac{e^{-\lambda_j t_i} - e^{-\lambda_j (t_i+\Delta t)}}{e^{-\lambda_j t_i}}} \qquad \triangleright \textit{ using Equation (1)} \\
=&\lim_{\Delta t \to 0} \frac{\frac{\lambda_1 e^{-\lambda_1 (t_1+\Delta t)}}{e^{-\lambda_1 t_1}}}{\sum_{j=1}^{d} \sum_{m_i \in \mathcal{T}_j} \frac{\lambda_j e^{-\lambda_j (t_i+\Delta t)}}{e^{-\lambda_j t_i}}} \qquad \triangleright \textit{ L'Hôpital's rule} \\
=&\lim_{\Delta t \to 0} \frac{\lambda_1 e^{-\lambda_1 \Delta t}}{\sum_{j=1}^{d} \sum_{m_i \in \mathcal{T}_j} \lambda_j e^{-\lambda_j \Delta t}} \\
=&\frac{\lambda_1}{\sum_{j=1}^{d} \sum_{m_i \in \mathcal{T}_j} \lambda_j} = \frac{\lambda_1}{\sum_{j=1}^{d} k_j \lambda_j} \qquad \triangleright\ |\mathcal{T}_j| = k_j
\end{aligned} \tag{2}$$

Note that the quantity $\Pr[m'_1 = m_1 | m_1 \in \mathcal{T}_1 \wedge O]$ is not equal to $\frac{1}{\sum_{j=1}^{d} k_j}$, and therefore, there is a bias for the message m'_1 to be a specific incoming message depending on the λ_j values. Intuitively, the messages with smaller delays are more likely to be from the type with smaller average delays. We also want to compute the probability that the message m'_1 is of type $\mathcal{T}_1$, and is computed as,

$$\begin{aligned}
\Pr[m'_1 \in \mathcal{T}_1 | O] &= \sum_{m_i \in \mathcal{T}_1} \Pr[m'_1 = m_i | m_i \in \mathcal{T}_1 \wedge O] \\
&= \frac{\lambda_1}{\sum_{j=1}^{d} k_j \lambda_j} \cdot k_1 \qquad \triangleright\ |\mathcal{T}_1| = k_1
\end{aligned} \tag{3}$$

Hereafter we drop the notation for the adversarial observations O for brevity, and assume all the probabilities are conditional to those observations.

3.2 Second and Subsequent Output Messages

After the first message m'_1 has left from the mixnode in the above example (c.f. Fig. 1), we want to calculate the probability of the next message m'_2 that is coming out of the mixnode is the target message $m_{\texttt{target}}$. However, now the adversary does not know the exact number of messages of each type inside the mixnode. Let us still assume that $m_1 = m_{\texttt{target}}$. There are three possible states for the mixnode after m'_1 has left:

- P0: m'_1 was actually m_1, which means that the mixnode does not contain m_1 anymore;
- P1: m'_1 was of type $\mathcal{T}_1$ but not m_1, and now there is one less message of type $\mathcal{T}_1$ contained in the mixnode;
- P2: m'_1 was of not of type $\mathcal{T}_1$ and the number of messages of type $\mathcal{T}_1$ contained in the mix remains the same.

For this subsection we assume (for simplicity) that there are only two types $\mathcal{T}_1$ and $\mathcal{T}_2$ of messages. We extend the analysis for

more than two types in Section 3.5. With the above assumption, the probability that the next message m'_2 is the specific input message m_1 can be calculated as:

$$\begin{aligned}
&\Pr[m'_2 = m_1 | m_1 \in \mathcal{T}_1] \\
&= \Pr[m'_2 = m_1 | P0 \wedge m_1 \in \mathcal{T}_1] \cdot \Pr[P0 | m_1 \in \mathcal{T}_1] \\
&\quad + \Pr[m'_2 = m_1 | P1 \wedge m_1 \in \mathcal{T}_1] \cdot \Pr[P1 | m_1 \in \mathcal{T}_1] \\
&\quad + \Pr[m'_2 = m_1 | P2 \wedge m_1 \in \mathcal{T}_1] \cdot \Pr[P2 | m_1 \in \mathcal{T}_1] \\
&= 0 \cdot \Pr[P0 | m_1 \in \mathcal{T}_1] + \frac{\lambda_1}{(k_1-1)\lambda_1 + k_2\lambda_2} \cdot \Pr[P1 | m_1 \in \mathcal{T}_1] \\
&\quad + \frac{\lambda_1}{k_1\lambda_1 + (k_2-1)\lambda_2} \cdot \Pr[P2 | m_1 \in \mathcal{T}_1]
\end{aligned}$$

Consequently, after i messages have left the mixnode, we need to consider all possible such combinations. Additionally, any new incoming message (after m'_1 has left) would also impact the probabilities corresponding to the later messages. Most importantly, the target message could arrive anytime (possibly after m'_1 has left). We need to consider all those possibilities to calculate the probabilities of the outgoing messages being the target message. So, given a specific target message m_{target} of type $\mathcal{T}_1$ (and it can arrive anytime during the protocol run), we keep track of the state of the mixnode with the following set of random variables:

- $G(j)$ denotes the probability that there are exactly j messages of type $\mathcal{T}_1$ inside the mixnode, and the target m_{target} has not yet arrived to the mixnode. We use $\mathcal{G}$ to denote the *event* that the target message has not arrived to the mixnode.
- $Q(j)$ denotes the event that there are exactly j messages of type $\mathcal{T}_1$ inside the mixnode and the target message is in the mixnode. We use $\mathcal{Q}$ to denote the event that target message has arrived and is still in the mixnode.
- $R(j)$ denotes the probability that there are exactly j messages of type $\mathcal{T}_1$ inside the mixnode, and the target message has left the mixnode. We use $\mathcal{R}$ to denote the event that the target message has left the mixnode.

The quantities $Q(j), G(j)$ and $R(j)$ are defined over $j \in [0, k]$ where $k = k_1 + k_2 - i$ denotes the total number of messages inside the mixnode; and i denotes the number of messages left the mixnode. Effectively, $G(j) = \Pr[\text{count}(\mathcal{T}_1) = j \wedge \mathcal{G}]$, and $\Pr[\mathcal{G}] = \sum_{j=0}^{k} G(j)$. Each of these quantities are updated when a new message arrives or a message leaves the mixnode. Additionally, as we will see shortly that $R(j)$ is maintained solely for the purpose of calculating the probability of an outgoing message being of type $\mathcal{T}_1$ or $\mathcal{T}_2$.

Initialization. We initialize $G(0) = 1, Q(0) = 0$ and $R(0) = 0$ before any messages arrive, since there are exactly 0 messages of type $\mathcal{T}_1$ inside the mixnode, and the target message has not yet arrived. This is consistent with the definitions of G, Q, and R.

3.2.1 When A Message Arrives. Until the target message arrives, $G(j)$ is updated for each $j \in [0, k+1]$ for each new incoming message m as follows:

$$G(j)^{new} = \begin{cases} G(j) & m \in \mathcal{T}_2 \\ G(j-1) & m \in \mathcal{T}_1 \wedge m \neq m_{\mathsf{target}}, j > 0 \\ 0 & m = m_{\mathsf{target}} \end{cases} \tag{4}$$

To explain briefly, whenever a message of type $\mathcal{T}_1$ arrives, the number of messages of type $\mathcal{T}_1$ in the mixnode goes from $(j-1)$ to j. If the mixnode had $(j-1)$ messages of type $\mathcal{T}_1$ with probability $G(j-1)$, now the mixnode has j messages with the same probability; and therefore, we have $G(j)^{new} = G(j-1)$. When the incoming message is of type $\mathcal{T}_2$, if the mixnode had j messages of type $\mathcal{T}_1$, the mixnode still has j messages of type $\mathcal{T}_1$; and therefore, the $G(j)$ remains unmodified. After the target message arrives, $G(j)$ becomes 0 for all $j \in [1, k]$, since the types of all incoming messages are known to the adversary.

Note that $G(j)^{new}$ (and $Q(j)^{new}$, $R(j)^{new}$ resp.) denotes the new value that will replace $G(j)$ (and $Q(j)$, $R(j)$ resp.) after the calculations are done for all the j values.

After the mixnode has received the target message, when a new message m arrives (including the target message itself), $Q(j)$ is updated for each $j \in [0, k+1]$ as follows:

$$Q(j)^{new} = \begin{cases} G(j-1) & m = m_{\mathsf{target}} \\ Q(j-1) & m \in \mathcal{T}_1, j > 0 \\ Q(j) & m \in \mathcal{T}_2 \end{cases} \tag{5}$$

Note that $Q(j)$ values are 0's for all for each $j \in [0, k+1]$ until the target message arrives to the mixnode. The main purpose of maintaining $G(j)$ values is to be able to correctly set the $Q(j)$ values when the target message arrives. And after the target message arrives, the update rules for $Q(j)$ are very similar to that of $G(j)$. However, very soon we are going to see that the $Q(j)$ and $G(j)$ values can be simultaneously non-zero when the adversary does not exactly know when the target message arrives (for a mixnode in the second and subsequent layers).

Analogous to $Q(j)$ values, $R(j)$ is updated for each $j \in [0, k+1]$ as follows:

$$R(j)^{new} = \begin{cases} R(j) & m \in \mathcal{T}_2 \\ R(j-1) & m \in \mathcal{T}_1, j > 0 \end{cases} \tag{6}$$

Note that, similar to $Q(j)$ values, we need to keep track of $R(j)$ values only after the target message has arrived to the mixnode. It is worth to mention here that $\sum_j Q(j)$ quantifies the probability that m_{target} is in the mixnode, whereas, $\sum_j G(j)$ quantifies the probability that m_{target} has not yet arrived to the mixnode. And, $\sum_j R(j)$ quantifies the probability that m_{target} has left the mixnode.

3.2.2 When A Message Leaves. Whenever a message m' leaves the mixnode, the probability that the message is the target message m_{target} can be calculated as (for a total number of k messages inside the mixnode before m' leaves),

$$\begin{aligned}
&\Pr[m' = m_{\mathsf{target}}] \\
&= \sum_{1 \le j \le k} \Pr[\mathcal{Q} \wedge \text{count}(\mathcal{T}_1) = j] \cdot \Pr[m' = m_{\mathsf{target}} \mid j = \text{count}(\mathcal{T}_1)] \\
&= \sum_{1 \le j \le k} Q(j) \cdot \frac{\lambda_1}{j\lambda_1 + (k-j)\lambda_2} \qquad \triangleright \textit{By Eq. (2)}
\end{aligned} \tag{7}$$

Explanation of Equation (7). Given that there are exactly j messages of type $\mathcal{T}_1$ and $(k-j)$ messages of type $\mathcal{T}_2$ held by the mixnode and the target message is one of those j messages, we know that

the probability of the next outgoing message being the target message can be calculated as $\frac{\lambda_1}{j\lambda_1 + (k-j)\lambda_2}$ (refer to Equation (2)). The probability that the mixnode has j messages of type $\mathcal{T}_1$ and the target is inside the mixnode is given by $Q(j)$. And, we have to consider the sum over all possible j values for which $Q(j)$ is non-zero. If the mixnode does not have the target message the next outgoing message cannot be the target message, and therefore, we do not need to consider the $G(j)$ or $R(j)$ values.

Other Probabilities of an Outgoing Message. We also want to compute the probability that the message m' is of type $\mathcal{T}_1$, and can be computed as,

$$\begin{aligned}&\Pr[m' \in \mathcal{T}_1]\\ &= \sum_{1\le j\le k} \Pr[\mathsf{count}(\mathcal{T}_1) = j] \cdot \Pr[m' \in \mathcal{T}_1 | j = \mathsf{count}(\mathcal{T}_1)]\\ &= \sum_{1\le j\le k} \big(G(j) + Q(j) + R(j)\big) \cdot \frac{j\lambda_1}{j\cdot\lambda_1 + (k-j)\lambda_2}\end{aligned} \tag{8}$$

Note that when the mixnode has j messages, either the target message has not yet arrived, or it is inside the mixnode, or it has left the mixnode. Therefore, the quantity $(G(j)+Q(j)+R(j))$ represents the probability of the mixnode holding exactly j messages of type $\mathcal{T}_1$. Consequently, $\sum_j G(j) + Q(j) + R(j) = 1$.

Similarly, the probability that the message m'_1 is of type $\mathcal{T}_2$ can be computed as,

$$\begin{aligned}&\Pr[m' \in \mathcal{T}_2]\\ &= \sum_{1\le j\le k} \Pr[\mathsf{count}(\mathcal{T}_1) = j] \cdot \Pr[m' \in \mathcal{T}_2 | j = \mathsf{count}(\mathcal{T}_1)]\\ &= \sum_{1\le j\le k} (G(j) + Q(j) + R(j)) \cdot \frac{(k-j)\lambda_2}{j\lambda_1 + (k-j)\lambda_2}\end{aligned} \tag{9}$$

Update G, Q, R When A Message Leaves. After the message m' leaves the mixnode, we also need to update $Q(j)$ and $G(j)$ values, and they are updated for each $j \in [0, k-1]$ as follows:

$$\begin{aligned}&Q(j)^{new}\\ &= Q(j) \cdot \Pr[m' \in \mathcal{T}_2 |\ j = \mathsf{count}(\mathcal{T}_1)]\\ &\quad + Q(j+1) \cdot \Pr[m' \in \mathcal{T}_1 \wedge m' \neq m_{\mathsf{target}} |\ j+1 = \mathsf{count}(\mathcal{T}_1)]\\ &= Q(j) \cdot \frac{(k-j)\lambda_2}{j\lambda_1 + (k-j)\lambda_2} + Q(j+1) \cdot \frac{j\lambda_1}{(j+1)\lambda_1 + (k-j-1)\lambda_2}\end{aligned} \tag{10}$$

For $j = k$, we update $Q(k)^{new} = 0$, since there are only $(k-1)$ messages left in the mixnode after m' has left. And,

$$\begin{aligned}G(j)^{new} &= G(j) \cdot \Pr[m' \in \mathcal{T}_2 |\ j = \mathsf{count}(\mathcal{T}_1)]\\ &\quad + G(j+1) \cdot \Pr[m' \in \mathcal{T}_1 |\ j+1 = \mathsf{count}(\mathcal{T}_1)]\\ &= G(j) \cdot \frac{(k-j)\lambda_2}{j\lambda_1 + (k-j)\lambda_2}\\ &\quad + G(j+1) \cdot \frac{(j+1)\lambda_1}{(j+1)\lambda_1 + (k-j-1)\lambda_2}\end{aligned} \tag{11}$$

For $j = k$, we update $G(k)^{new} = 0$. Additionally,

$$\begin{aligned}R(j)^{new} &= R(j) \cdot \Pr[m \in \mathcal{T}_2 |\ j = \mathsf{count}(\mathcal{T}_1)]\\ &\quad + R(j+1) \cdot \Pr[m \in \mathcal{T}_1 |\ j+1 = \mathsf{count}(\mathcal{T}_1)]\\ &\quad + Q(j+1) \cdot \Pr[m' = m_{\mathsf{target}} |\ j+1 = \mathsf{count}(\mathcal{T}_1)]\\ &= R(j) \cdot \frac{(k-j)\lambda_2}{j\lambda_1 + (k-j)\lambda_2}\\ &\quad + R(j+1) \cdot \frac{(j+1)\lambda_1}{(j+1)\lambda_1 + (k-j-1)\lambda_2}\\ &\quad + Q(j+1) \cdot \frac{\lambda_1}{(j+1)\lambda_1 + (k-j-1)\lambda_2}\end{aligned} \tag{12}$$

For $j = k$, we update $R(k)^{new} = 0$.

3.3 General Case With Two Traffic Types

For a mixnode on the second and consequent layers of a mixnet, the adversary might not exactly know the types of the incoming messages to the mixnode. However, based on the observed delays on the previous layers (and following the analysis in Section 3.2), the adversary can guess the type of each message with some probability. With that consideration, we want to analyze how easily the adversary can correlate the outgoing messages with the incoming messages. For example, for a mixnode on the second layer the adversary can compute the probabilities of each incoming message being type $\mathcal{T}_1$ (as shown in Section 3.2), and each of them will have a probability of being the target message (we are still assuming that the target message is a message from the traffic type $\mathcal{T}_1$). In such cases, we want to compute the probabilities of the outgoing messages of being the target message.

Similar to Section 3.2, we still assume that there are only two types of messages: $\mathcal{T}_1$ and $\mathcal{T}_2$. We keep track of the state of the mixnode using the variables $Q(j)$, $G(j)$ and $R(j)$ for $j \in [0, k]$ where k denotes the total number of message held by the mixnode. For a mixnode in the first layer, the adversary knows the exact number k_1 (resp. k_2) of messages of type $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) came to the mixnode. However, for a mixnode in the second layer, those quantities are probabilistic and dependent on the first layer. Additionally, the adversary does not know when the target message arrives to the mixnode, if at all (since there can many mixnodes in every layer).

3.3.1 Update G, Q, R When A Message Arrives. Before any messages arrive we initialize $G(0) = 1, Q(0) = 0$ and $R(0) = 0$. When a new message m arrives, $G(j)$ can be updated for each $j \in [1, k+1]$ as:

$$\begin{aligned}&G(j)^{new}\\ &= \Pr[m \in \mathcal{T}_2 \wedge \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j]\\ &\quad + \Pr[m \in \mathcal{T}_1 \wedge m \neq m_{\mathsf{target}} \wedge \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j-1]\\ &= \Pr[m \in \mathcal{T}_2 \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j] \times G(j)\\ &\quad + \Pr[m \in \mathcal{T}_1 \wedge m \neq m_{\mathsf{target}} \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j-1] \times G(j-1)\\ &= \Pr[m \in \mathcal{T}_2 \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j] \times G(j)\\ &\quad + \big(\Pr[m \in \mathcal{T}_1 \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j-1]\\ &\quad - \Pr[m = m_{\mathsf{target}} | \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j-1]\big) \times G(j-1)\end{aligned} \tag{13}$$

And for $j = 0$ we can update

$$G(0)^{new} = G(0) \times \Pr[m \in \mathcal{T}_2 \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j].$$

Note that the number of messages of type $\mathcal{T}_1$ in the mixnode on the second (or subsequent) layer depends on the state of the previous layer(s). In that sense, $\Pr[m \in \mathcal{T}_2 \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j]$ depends on the state of the mixnode, which in turn depends on the state of previous layers. We consider the following approximation: the probability of an incoming message being of type $\mathcal{T}_1$ (resp. type $\mathcal{T}_2$) is independent of $\mathsf{count}(\mathcal{T}_1)$ of the current mixnode; and therefore, $\Pr[m \in \mathcal{T}_1 \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j] = \Pr[m \in \mathcal{T}_1 \mid \mathcal{G}] = \Pr[m \in \mathcal{T}_1]$. Similarly, $\Pr[m \in \mathcal{T}_2 \mid \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j] = \Pr[m \in \mathcal{T}_2]$. Note that the type of the incoming message m is always independent of where the target message is, however, $\Pr[m = m_{\mathsf{target}}]$ is not. So, we have the following,

$$G(j)^{new} = \Pr[m \in \mathcal{T}_2] \times G(j) + \left(\Pr[m \in \mathcal{T}_1] - \frac{\Pr[m = m_{\mathsf{target}}]}{\Pr[\mathcal{G}]} \right) \times G(j-1). \tag{14}$$

Note that we have used $\Pr[m = m_{\mathsf{target}} | \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j-1] = \frac{\Pr[m = m_{\mathsf{target}}]}{\Pr[\mathcal{G}]}$ in the final equation. That is because the target message can be present only with one mixnode. Therefore, m could only be a target message if it has not arrived to the current mixnode before, or in other words, if $\mathcal{G}$ is true. Consequently,

$$\Pr[m = m_{\mathsf{target}} \wedge \mathcal{G}] = \Pr[m = m_{\mathsf{target}}]$$
$$\iff \Pr[m = m_{\mathsf{target}} | \mathcal{G}] = \frac{\Pr[m = m_{\mathsf{target}}]}{\Pr[\mathcal{G}]}$$

Justification for the Approximation. The justification is actually two-fold: (1) In a steady-state flow, based on the analysis from [4], it is a good approximation to consider each mixnode as an independent $M/M/\infty$ queue. That means, the internal states of the nodes can be considered independent of each other; except for the adversarial knowledge of the total number of messages contained in a node. With that approximation, the probability that an incoming message from the previous layer is of type $\mathcal{T}_1$ or $\mathcal{T}_2$ is independent of the state of current mixnode. However, the adversary has the knowledge that the target message can be held by only one mixnode at any given point. And therefore, $\Pr[m = m_{\mathsf{target}}]$ for an incoming message m is not independent of the state of the current mixnode, and is bounded by $\Pr[\mathcal{G}]$. (2) If we want to plug-in these probability calculations in a simulator (which we do in Section 4), maintaining the inter-dependent states for a mixnet with multiple layers and multiple mixnodes per layer explodes the computational complexity, and becomes a serious performance bottleneck.

$Q(j)$ is updated for each $j \in [1, k+1]$ as:

$$\begin{aligned} Q(j)^{new} = {} & \Pr[m \in \mathcal{T}_2 \wedge Q \wedge \mathsf{count}(\mathcal{T}_1) = j] \\ & + \Pr[m \in \mathcal{T}_1 \wedge m \neq m_{\mathsf{target}} \wedge Q \wedge \mathsf{count}(\mathcal{T}_1) = j-1] \\ & + \Pr[m = m_{\mathsf{target}} \wedge \mathcal{G} \wedge \mathsf{count}(\mathcal{T}_1) = j-1] \\ = {} & Q(j) \cdot \Pr[m \in \mathcal{T}_2] + Q(j-1) \cdot \Pr[m \in \mathcal{T}_1] \\ & + G(j-1) \cdot \frac{\Pr[m = m_{\mathsf{target}}]}{\Pr[\mathcal{G}]} \end{aligned} \tag{15}$$

And for $j = 0$, we update $Q(0)^{new} = Q(0) \cdot \Pr[m \in \mathcal{T}_2]$.

Explanation for Equations (14) and (15) in Simple Words. The concept of transition between G and Q is still similar to the previous section (Section 3.2). However, now the transition is probabilistic, since the adversary does not certainly know if an incoming message m is the target message. If m has a probability P of being the target message, overall $\Pr[Q^{new}]$ should be equal to $\Pr[Q]$ + P; in other words, $\sum Q(j)^{new} = \sum Q(j) + P$.

For each j, $G(j)^{new}$ is contributed by $G(j)$ with an amount exactly as the probability that m belongs to $\mathcal{T}_2$, and $G(j-1)$ with an amount exactly as the probability that m belongs to $\mathcal{T}_1$ but not the target. So, in total G quantities are reduced by an amount same as the probability of m being the target. And, that amount is added to the total of Q quantities by adding the amounts that we have just reduced from G values to each $Q(j)^{new}$. So, for each j, $Q(j)^{new}$ is contributed by $Q(j)$ (with an amount exactly as the probability that m belongs to $\mathcal{T}_2$), $Q(j-1)$ (with an amount exactly as the probability that m belongs to $\mathcal{T}_1$), and the amount laid off from $G(j-1)$.

Note that $G(j)^{new}$ (resp. $Q(j)^{new}$) is contributed by $G(j)$ (resp. $Q(j)$) for the amount as the probability that m belongs to $\mathcal{T}_2$ because the number of messages of $\mathcal{T}_1$ remains the same when $m \in \mathcal{T}_2$. Analogously, $G(j)^{new}$ (resp. $Q(j)^{new}$) is contributed by $G(j-1)$ because the number of messages of $\mathcal{T}_1$ increases by 1 when $m \in \mathcal{T}_1$.

Analogously, $R(j)$ is updated for each $j \in [1, k+1]$ as follows:

$$\begin{aligned} R(j)^{new} = {} & R(j) \cdot \Pr[m \in \mathcal{T}_2] \\ & + R(j-1) \cdot \Pr[m \in \mathcal{T}_1 \wedge m \neq m_{\mathsf{target}}] \end{aligned} \tag{16}$$

And for $j = 0$, we update $R(0)^{new} = R(0) \cdot \Pr[m \in \mathcal{T}_2]$.

3.3.2 When A Message Leaves. When a message leaves from the mixnode we update the quantities $G(j), Q(j), R(j)$ for $j \in [0, k]$ exactly same as in Section 3.2. The probability that an outgoing message is the target message, the probability that it belongs to a specific type (e.g., $\mathcal{T}_1$) are also computed in the same way as in Section 3.2.

We present the overall methodology to calculate the probabilities for a given mixnode as part of a simulator in Algorithm 1. It is worth to mention here that our method also works when there is only one type of traffic going through the mixnet, and provides the same results as the existing methods [15, 23] for a single traffic type.

Target Is From Type $\mathcal{T}_2$. When the target message is from $\mathcal{T}_2$ the derivation remain exactly the same, however, all the quantities are to be defined for type $\mathcal{T}_2$ — which is equivalent to swapping the assignment of the types in the notation.

3.4 When the Recipient Leaks the Types

If the mixnode is the last layer and the recipient leaks the type of the message it is receiving, the adversary gains additional knowledge. In such cases, the probabilities need to be adjusted to consider that factor. Suppose, the outgoing messages from the mixnode are denoted with $m'_1, m'_2, \ldots$ etc. And the probabilities of them being the target message m_{target} are $p_1, p_2, \ldots$ respectively, without using the knowledge from the recipient side. Let us consider that $m_{\mathsf{target}} \in \mathcal{T}_1$. With the additional knowledge from the recipient

side, we can derive the following for an outgoing message $m'_i \in \mathcal{T}_1$,

$$\begin{aligned}&\Pr[m'_i = m_{\mathsf{target}} \mid m'_i \in \mathcal{T}_1 \wedge m_{\mathsf{target}} \in \mathcal{T}_1]\\ &= \frac{\Pr[m'_i = m_{\mathsf{target}} \wedge m'_i \in \mathcal{T}_1 \mid m_{\mathsf{target}} \in \mathcal{T}_1]}{\Pr[m'_i \in \mathcal{T}_1 \mid m_{\mathsf{target}} \in \mathcal{T}_1]}\\ &= \frac{\Pr[m'_i = m_{\mathsf{target}} \wedge m'_i \in \mathcal{T}_1 \mid m_{\mathsf{target}} \in \mathcal{T}_1]}{\Pr[m'_i \in \mathcal{T}_1]} \;=\; \frac{p_i}{D_1};\end{aligned}$$

where $D_1 = \sum_{i:m'_i \in \mathcal{T}_1} p_i$. Similarly, for an outgoing message $m'_i \in \mathcal{T}_2$ we can derive,

$$\begin{aligned}&\Pr[m'_i = m_{\mathsf{target}} \mid m'_i \in \mathcal{T}_2 \wedge m_{\mathsf{target}} \in \mathcal{T}_1]\\ &= \frac{\Pr[m'_i = m_{\mathsf{target}} \wedge m'_i \in \mathcal{T}_2 \mid m_{\mathsf{target}} \in \mathcal{T}_1]}{\Pr[m'_i \in \mathcal{T}_2 \mid m_{\mathsf{target}} \in \mathcal{T}_1]} \;=\; 0.\end{aligned}$$

Therefore, we can adjust the probabilities for the leakage on the recipient side by normalizing them for all the outgoing messages of type $\mathcal{T}_1$ (for the target message $m_{\mathsf{target}} \in \mathcal{T}_1$).

3.5 More Than Two Types of Traffic

The methodology presented until now can be extended to analyze anonymity when there are more than two types of traffic. However, an additional set of quantities would be required to keep track of the number of messages for each type inside the mixnode, except the type of the target.[2] However, we do not need to modify G, Q, R calculations, because they only concern about if an incoming or outgoing message is of the same type as the target or not. The new quantities corresponding to every type are very similar to e.g., G with a slight difference — they are not conditional on the arrival (or departure) of the target message. How they are updated when a message arrives or leaves are also similar.

Assuming that the target is from type $\mathcal{T}_1$, let $H_w(j)$ denote the probability that there are j messages inside the mixnode of type $\mathcal{T}_w$ except $w = 1$; and,

$$H_w(j|j < x) = \begin{cases} \dfrac{H_w(j)}{\sum_{a=0}^{x} H_w(a)} & j < x \\ 0 & \text{otherwise} \end{cases}$$

denote the probability that there are j messages inside the mixnode of type $\mathcal{T}_t$ but conditioned on $j < x$. Note that $H_1(j) = G(j) + Q(j) + R(j)$. Once the H_w quantities are in place, the probabilities of an outgoing message being a target message, or of a specific type can be computed similar to Section 3.3. For example, the probability of an outgoing message being the target message can be calculated as (assuming a total of d types),

$$\begin{aligned}&\Pr[m' = m_{\mathsf{target}}]\\ &= \sum_{1\le k_d\le k} \cdots \sum_{1\le k_1\le k} H_d(k_d)\cdots H_2(k_2)\times Q(k_1)\\ &\qquad \times \Pr[m' = m_{\mathsf{target}} \mid k_t = \mathsf{count}(\mathcal{T}_t)\ \forall 1 \le t \le k_d]\\ &= \sum_{k_d=0}^{k} \cdots \sum_{k_1=1}^{k} H_d(k_d|k_d \le k - k_{d-1} - \cdots - k_1)\cdots\\ &\qquad \times H_2(k_2|k_2 \le k - k_1)\times Q(k_1)\times \frac{\lambda_1}{\sum_{a=1}^{d} k_a\lambda_a}\end{aligned} \tag{17}$$

However, the H_w quantities need to be updated whenever a message arrives or leaves. Whenever a new message arrives, $H_t(j)$ values can be updated as follows,

$$H_w(j)^{new} = H_w(j)\cdot \Pr[m \in \mathcal{T}_w] + H_w(j-1)\cdot \Pr[m \notin \mathcal{T}_w];$$

where $\Pr[m \in \mathcal{T}_w]$ and $\Pr[m \notin \mathcal{T}_w]$ are calculated based on the previous layer. And when a message leaves,

$$\begin{aligned}H_w(j)^{new} &= H_w(j)\cdot \Pr[m \notin \mathcal{T}_w|\ j = \mathsf{count}(\mathcal{T}_w)]\\ &\quad + H_w(j+1)\cdot \Pr[m \in \mathcal{T}_w|\ j+1 = \mathsf{count}(\mathcal{T}_w)]\end{aligned} \tag{18}$$

For $w = 1$ we can say,

$$\begin{aligned}&\Pr[m \in \mathcal{T}_w|\ j = \mathsf{count}(\mathcal{T}_w)]\\ &= \sum_{k_d=0}^{k-j} \cdots \sum_{k_2=1}^{k-j} H_d(k_d|k_d \le k - k_{d-1} - \cdots - k_2 - j)\cdots\\ &\qquad \times H_2(k_2|k_2 \le k-j)\times \frac{j\lambda_w}{\sum_{a=2}^{d} k_a\lambda_a + j\lambda_w}\end{aligned} \tag{19}$$

and $\Pr[m \notin \mathcal{T}_w|\ j = \mathsf{count}(\mathcal{T}_w)] = 1 - \Pr[m \in \mathcal{T}_w|\ j = \mathsf{count}(\mathcal{T}_w)]$. The evaluation is exactly the same for any other w, except for the switched indices of the variables.

4 END-TO-END ANONYMITY ANALYSIS FOR MIXNETS WITH BETA-MIXING

In order to demonstrate the effect of blending, we provide empirical analysis for end-to-end mixnets with the simple case of two types of traffic, and discuss in Section 4.6 how to extend the insights from these analysis when there are more types.

4.1 Methodology

Our methodology to evaluate the impact of blending different traffic types on top of the same mixnet is as follows: First, we presented our analytical method in the previous section which shows how to calculate the probability of an output message being one input message. We then modified the open-source simulator used in [3] by implementing our analytical method. The modifications made to the simulator are summarized in Algorithm 1. We executed the updated simulator across various mixnet configurations (number of nodes, number of layers, rates of the different traffic generations etc). Finally, as a measure of anonymity, we evaluate the entropy of the probability distribution linking the mixnet's input and output messages [15, 23]. For our evaluations, we assume that the adversary knows the types of the messages coming to the mixnet. Note that, while the traffic type is not immediately visible, it may be possible

[2] With only two types, the number of messages of type $\mathcal{T}_2$ can easily be calculated if the number of messages of $\mathcal{T}_1$ and the total number of messages are known. However, that is not the case when there are many types.

to infer it, for example if the message sending rate is indicative of which application may be generating the traffic.

Algorithm 1: Probability computation on a single node with two types of traffic.

Result: Updated $\Pr[m_i = m_t | m_t \in \mathcal{T}_1]$.
Initialize:
G, Q, R : arbitrarily expandable Lists ;
$G.append(1)$; $Q.append(0)$;
$k = 0$; $i = 0$;
if *event(receive(m_i))* **then**
  $k++$;
  for $j \leftarrow 0$ **to** k **do**

$$Q[j] = Q[j-1] \cdot \Pr[m_i \in \mathcal{T}_1] + Q[j] \cdot \Pr[m_i \in \mathcal{T}_2] + G[j-1] \cdot \frac{\Pr[m_i = m_t]}{\text{sum}(G)};$$

$$G[j] = G[j] \cdot \Pr[m_i \in \mathcal{T}_2] + G[j-1] \cdot \left(\Pr[m_i \in \mathcal{T}_1] - \frac{\Pr[m_i = m_t]}{\text{sum}(G)}\right);$$

$$R[j] = R[j-1] \cdot \Pr[m_i \in \mathcal{T}_1] + R[j] \cdot \Pr[m_i \in \mathcal{T}_2];$$

  end
end
if *event(send(m_i))* **then**
  define $\text{denom}(j) = j \cdot \lambda_1 + (k - i - j) \cdot \lambda_2$;

$$\Pr[m_i = m_t] = \sum_{j=0}^{k-1} \frac{\lambda_1}{\text{denom}(j)} \cdot Q[j];$$

$$\Pr[m_i \in \mathcal{T}_1] = \sum_{j=0}^{k-1} \frac{j \cdot \lambda_1}{\text{denom(j)}} \cdot (Q[j] + R[j] + G[j]);$$

$$\Pr[m_i \in \mathcal{T}_2] = \sum_{j=0}^{k-1} \frac{(k-i-j) \cdot \lambda_2}{\text{denom}(j)} \cdot (Q[j] + R[j] + G[j]);$$

  for $j \leftarrow 0$ **to** k **do**

$$R[j] = \frac{\lambda_1}{\text{denom}(j+1)} \cdot Q[j+1] + \frac{(j+1) \cdot \lambda_1}{\text{denom}(j+1)} \cdot R[j+1] + \frac{(k-i-j) \cdot \lambda_2}{\text{denom}(j)} \cdot R[j];$$

$$Q[j] = \frac{j \cdot \lambda_1}{\text{denom}(j+1)} \cdot Q[j+1] + \frac{(k-i-j) \cdot \lambda_2}{\text{denom}(j)} \cdot Q[j];$$

$$G[j] = \frac{(j+1) \cdot \lambda_1}{\text{denom}(j+1)} \cdot G[j+1] + \frac{(k-i-j) \cdot \lambda_2}{\text{denom}(j)} \cdot G[j];$$

  end
  Forward Message (m_i);
  $i++$;
end

4.2 Experimental Setup

The simulation starts with users generating messages, selecting a route for each message, and sending them through the network to their respective recipients. Each user generates messages from two traffic types, $\mathcal{T}_1$ and $\mathcal{T}_2$, following a Poisson distribution with parameters λ'_1 and λ'_2. The delays parameters of each message belonging to either $\mathcal{T}_1$ or $\mathcal{T}_2$ also follow Poisson Distribution with parameters λ_1 and λ_2, respectively. We consider a user population of $\mathcal{U} = 100$ users, each user generating a total of 5 messages per second for the two types of traffic combined. The total traffic generation rate is 500 messages per second with a total of 100000 messages. For each simulation run, each run representing one data point in the graphs, once the network has been initialized and is in a steady state we choose 50 input target messages[3]. In order to evaluate the impact of blending two types of traffic on the anonymity of the system for each of these types of traffic, we choose target messages from each type of traffic. $m_{\texttt{target}}$ is the message that the adversary follows. At the end of the simulations, all the 100000 received messages, in one simulation run, have a probability of being $m_{\texttt{target}}$. Finally we plug this probability distribution in the entropy metric in order to evaluate the anonymity provided by the mixnet.

We vary for different experiments the generation rates of each traffic such that the sum of the two rates of traffic generations $(\lambda'_1 + \lambda'_2)$ is equal to 5. Our goal is to evaluate whether there are advantages of blending two different traffic types with different latency requirements in the same mixnet over maintaining separate mixnet infrastructures as well as determining the significance of the traffic generation ratios (λ'_1:λ'_2). We consider $\mathcal{T}_1$ as `fast` traffic, and $\mathcal{T}_2$ as `slow` traffic, meaning all messages belonging to $\mathcal{T}_1$ have an average delay $\frac{1}{\lambda_1} = d_1 = 1$, and messages from $\mathcal{T}_2$ have higher delays. For each experiment, we plot the entropy values, for two cases: (i) for target messages belonging to $\mathcal{T}_1$ and (ii) for messages belonging to $\mathcal{T}_2$. To evaluate the impact of blending traffic on the anonymity provided by the mixnet for messages from:

- $\mathcal{T}_1$ (target messages are from type $\mathcal{T}_1$): The first 4 data points represent the entropy values for : ($\lambda'_1 = 5, \lambda'_2 = 0$), ($\lambda'_1 = 4, \lambda'_2 = 1$), ($\lambda'_1 = 2.5, \lambda'_2 = 2.5$), and ($\lambda'_1 = 1$, $\lambda'_2 = 4$) The 5th data point of this graph represent only one target message from $\mathcal{T}_1$ and the rest of the network traffic is from $\mathcal{T}_2$ with $\lambda'_2 = 5$.
- $\mathcal{T}_2$ (target messages are from type $\mathcal{T}_2$): The first 4 data points represent the entropy values for: ($\lambda'_1 = 0, \lambda'_2 = 5$), ($\lambda'_1 = 1, \lambda'_2 = 4$), ($\lambda'_1 = 2.5, \lambda'_2 = 2.5$), and ($\lambda'_1 = 4$, $\lambda'_2 = 1$). The 5th data point of this graph represent only one target message from $\mathcal{T}_2$ and the rest of the network traffic is from $\mathcal{T}_1$ with $\lambda'_1 = 5$.

In order to compare anonymity of the messages of $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) where traffic are blended to the anonymity where there's a dedicated infrastructure for each traffic type, we also plot the entropy values for the same values of λ'_1 (resp. λ'_2) but all values of λ'_2 (resp. λ'_1) are equal to 0. We call the scenario a `Solo` case, meaning that the network only has messages from the the type traffic $\mathcal{T}_1$ ($\mathcal{T}_2$). Finally, in the scenario of one single message from either $\mathcal{T}_1$ or $\mathcal{T}_2$, we want to evaluate the anonymity provided by the network when there's only one message from that type of traffic and the rest of the 500 messages per second are from the opposite traffic. Such scenarios can manifest in real-world situations. For instance, in the case of Nym [12], a practical scenario might involve one user initiating a cryptocurrency transaction, while the rest of the network during a rather large period of time, are sending Telegram messages. We consider the following mixnet settings for our evaluations:

(1) `Cascade`: One mixnode per layer for $L = 1$, $L = 2$ and $L = 3$ (§4.3);
(2) `3x10`: A mixnet consisting of 3 layers with 10 mixnodes per layer (§4.4):

[3]The simulation times vary for experiments due to the extensive probability computation for each output message. We will include the open-source code, the data, as well as other details related to the simulator as an artefact.

- against a Global Passive Adversary (GPA);
- the GPA additionally compromises 10% mixnodes;
- the GPA can see the types of all output messages;

(3) $d_1 : d_2$: Different ratios of per-mix delays; $\mathcal{T}_1$ with an average delay d_1=1 and $\mathcal{T}_2$ with average delays $d_2 = 5$, $d_2 = 10$, $d_2 = 15$ and $d_2 = 20$ (§4.5);

4.3 Evaluation: One Mixnode Per Layer

First, we evaluate the entropy for a small mixnet: one mixnode per layer for $L = 1$ (one single standalone mixnode), $L = 2$ and for $L = 3$. We plot the entropy values for the different λ'_1 and λ'_2 ratios evaluating the anonymity provided by the mixnet for messages from $\mathcal{T}_1$ in Figure 2a and from $\mathcal{T}_2$ in Figure 2b.

In these figures, the blue lines denote the entropy values for 1 single mixnode, the red ones denote the entropy for $L = 2$ and the black is for $L = 3$. The solid lines for each of these configuration denote the entropy values of the two types of traffic blended together and the dashed lines are for the `Solo` cases, meaning that all λ' values from the opposite traffic are equal to 0.

As we can see in Figure 2a, as the ratio $\lambda'_1 : \lambda'_2$ declines when blending traffic (solid lines), the entropy only slightly decreases. The `Solo` cases, represented by the dashed lines, show the entropy values for single traffic $\mathcal{T}_1$ in the mixnet. The decreasing entropy values of the `Solo` cases are to be expected since there is a decrease in the traffic generation rate λ'_1 in 2a. We use the `Solo` cases as reference in order to compare the impact of blending on anonymity to dedicating a different mixnet per traffic type. We conclude from this graph that even though there's a slight decrease, the overall anonymity is much better when blending traffic. This is due to the fact that messages from $\mathcal{T}_2$ do make up for the reduced number of messages of $\mathcal{T}_1$.

Figure 2b provides similar observations. Additionally, it shows that the slow traffic ($\mathcal{T}_2$) has much better entropy compared to the fast traffic ($\mathcal{T}_1$) for both `Solo` and with blending. This is due to the higher per-mix delay ($d_2 = 5$) for messages from $\mathcal{T}_2$ compared to messages from $\mathcal{T}_1$ ($d_1 = 1$). However we do notice a slight increase in anonymity for traffic type $\mathcal{T}_2$ when the ratios $\lambda'_2 : \lambda'_1$ declines. This is due to the fact that messages $\mathcal{T}_2$ are able to meet with many more messages of fast traffic ($\mathcal{T}_1$) when they are inside the mix because of their higher delays. In other words, for a message with a large delay, having other messages with small delays decrease the probability of that message being one input message and hence this provide higher value of entropy.

When comparing the entropy values for a single mixnode to the case of $L = 2$ and $L = 3$, we notice that adding layers increase anonymity, a result that is consistent with previous works [3, 17, 22] analysing only a single type of traffic. For example, when having $L = 1$ and $W = 1$ (a single standalone mixnode represented by the blue lines), we have entropy values of around 10.3 for type $\mathcal{T}_1$ and $\lambda'_1 = 1, \lambda'_2 = 4$; and entropy values of around 11.5 for the same λ' when $L = 3$ and $W = 1$ (black lines).

However, when there's only one single unique target message, we notice that adding layers has the opposite impacts, especially for the slow traffic. This is due to the fact that the adversary has more advantages in following this unique target message that has a different delay compared to all ther other messages in the network. We emphasize however the importance of blending, because otherwise this *unlucky* message would be 100% de-anonymized as shown by the dashed lines representing the `solo` cases.

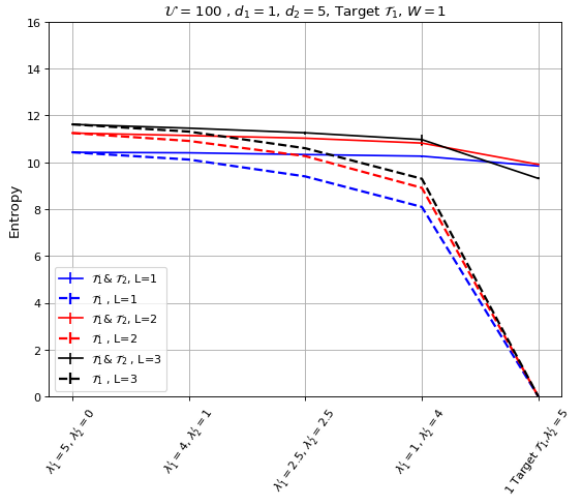
We emphasize the validity of our analysis by comparing our simulation results in `Solo` cases with those from [17]. This comparison is justified as the authors in [17] exclusively evaluated one type of traffic, and both methods yield identical entropy values when utilizing the same network size, user population, and traffic generation parameters.

4.4 Evaluation: 3 Layers, 10 Mixnodes Per Layer

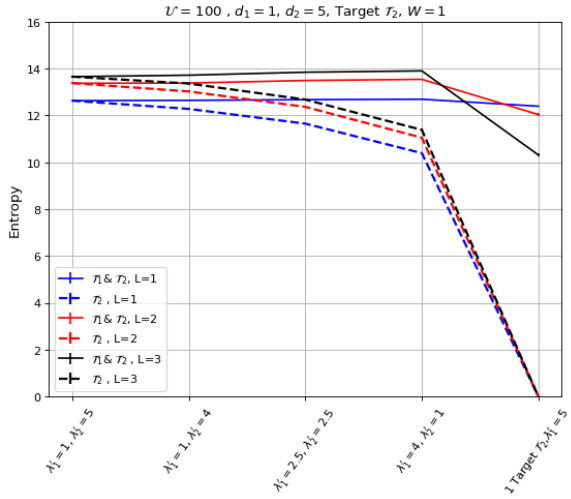
In this section, we evaluate the impact of blending the two types of traffic in a network with $L = 3$ and $W = 10$. We maintain the same experimental setups, in terms of number of clients and number of messages per second, as in the previous experiments. Additionally, we evaluate the impact of having an adversary who is corrupting 10% of the network ($B = 3$) as well as the impact of an adversary that additionally knows the traffic type of each output message. We symbolise by $\mathcal{A}$ the adversary who only knows the types of input messages and by $\mathcal{B}$ the adversary who knows the the types of traffic of all input and output messages. The type of messages when received by the recipient can be leaked when the adversary can compromise the recipient, or controls the ISP of the recipient. In order to quantify this knowledge of the adversary in our empirical analysis, we normalize the probabilities of all messages being the target as explained in Section 3.4. We report the entropy values in Figure 3.

In Figure 3, the solid lines represent the entropy values when the two types of traffic are blended and the dashed line represent the `Solo cases`. The red lines represent the entropy values for 0 corruption, the blue lines represent the adversary who corrupts 10% of all the nodes ($B = 3$) and the black lines represent the scenario $\mathcal{B}$ where the adversary knows the types of traffic of all the input and output messages. Similar to previous experiments, we observe that the entropy decreases as λ'_1 (resp. λ'_2) decreases when traffic types are not blended; however, when we blend messages from different traffic types, the messages from the second traffic type compensates for the lack of messages from $\mathcal{T}_1$ (resp. $\mathcal{T}_2$). Similar to previous evaluation, the slow traffic benefits slightly more from blending than the fast traffic.

When we consider 10% of the mixnodes are compromised (we choose them by choosing one corrupt mixnode per layer), we observe slight decrease in entropy values for each type of traffic, and in both blended and non-blended scenario (c.f. Figure 3). The black lines, representing adversary $\mathcal{B}$, shows that the entropy is lower when the adversary knows the types of output messages than when the adversary does not know — which was expected, since the adversary has the additional information of the output traffic types. In fact, the entropy values closely match the dashed red lines depicting entropy of single-type traffic. This indicates that, even in the worst-case scenario where the adversary knows the types of traffic of all input and output messages, blending traffic is as effective as dedicating the entire network to each traffic type. We can summarize the insights as follows:

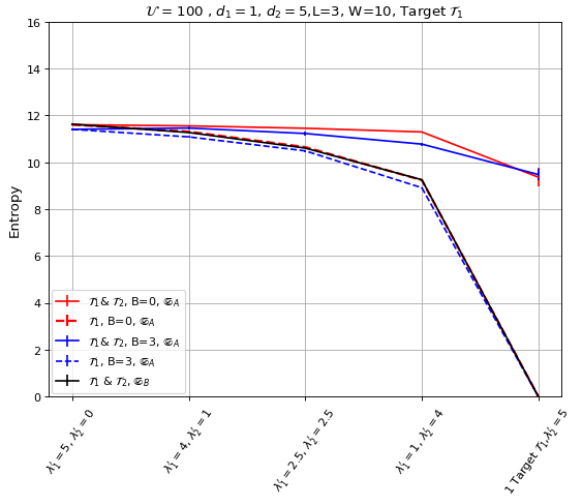


(a) Entropy for messages of type $\mathcal{T}_1$ for $W = 1$, $L = 1$, $L = 2$ and $L = 3$.

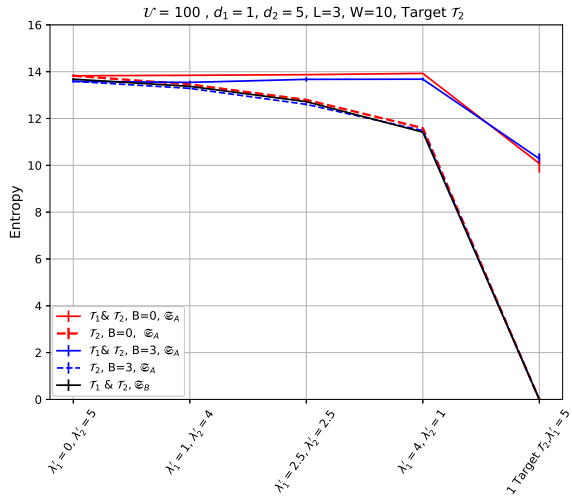


(b) Entropy for messages of type $\mathcal{T}_2$ for $W = 1$, $L = 1$, $L = 2$ and $L = 3$.

Figure 2: Evaluation of anonymity in terms of entropy for continuous mixnets with width $W = 1$ (number of mixnodes per layer), average delay $d_1 = \frac{1}{\lambda_1} = 1$ for traffic type $\mathcal{T}_1$, average delay $d_2 = \frac{1}{\lambda_2} = 5$ for traffic type $\mathcal{T}_2$.



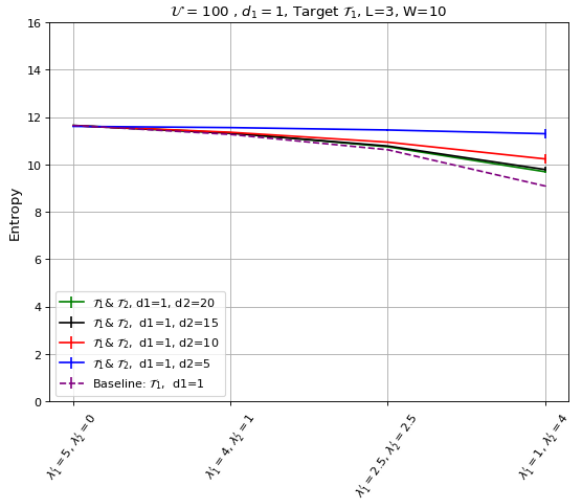
(a) Entropy for messages of type $\mathcal{T}_1$ for a network with $L = 3$, and $W = 10$ and two adversarial setups.



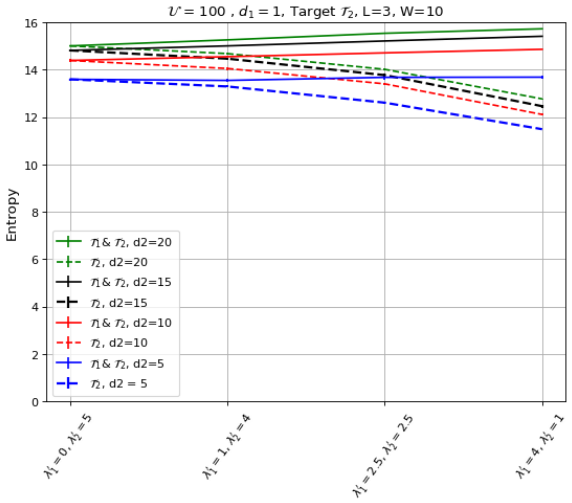
(b) Entropy for messages of type $\mathcal{T}_2$ for a network with $L = 3$, and $W = 10$ and two adversarial setups.

Figure 3: Evaluation of anonymity in terms of entropy for continuous mixnets with width $W = 10$ (number of mixnodes per layer), number of layers $L = 3$, average delay $d_1 = \frac{1}{\lambda_1} = 1$ for traffic type $\mathcal{T}_1$, average delay $d_2 = \frac{1}{\lambda_2} = 5$ for traffic type $\mathcal{T}_2$. B is the number of compromised mixnodes in the network. Scenario $\mathcal{A}$: Adversary only knows the input types of traffic. Scenario $\mathcal{B}$: Adversary knows the type of traffic of input and output messages

(1) Blending traffic is slightly more advantageous for the slow traffic than for the fast traffic.
(2) However even for the fast traffic, blending two different types of traffic into the same mixnet infrastructure is more beneficial in terms of anonymity compared to sending them through separate mixnets.
(3) On the other hand, anonymity for the slow traffic improves as the difference between the delay parameters increases.



(a) Entropy for messages of type $\mathcal{T}_1$ for $L = 3$, $W = 10$, $B = 0$.



(b) Entropy for messages of type $\mathcal{T}_2$ for $L = 3$, $W = 10$, $B = 0$.

Figure 4: Evaluation of anonymity in terms of entropy for continuous mixnets with $L = 3$, $W = 10$, average delay $d_1 = \frac{1}{\lambda_1} = 1$ for traffic type $\mathcal{T}_1$, and different average delays for traffic type $\mathcal{T}_2$: $d_2 = \frac{1}{\lambda_2} = 5$, $d_2 = 10$, $d_2 = 15$, $d_2 = 20$.

4.5 Evaluation: Different Ratios of Delay Parameters

In the previous sections, we consider two types of traffic which we called *fast* with an average delay $d_1 = 1$ and a second type of traffic $\mathcal{T}_2$ which we called *slow* with an average delay $d_2 = 5$. In this section we want to investigate the impact of these per mix delays ratios on the anonymity of both of the traffic. In figure 4a, we keep the same rate for per-mix delay $d_1 = 1$ for traffic type $\mathcal{T}_1$, and we change the rate delays of traffic $\mathcal{T}_2$ to $d_2 = 10$ (red), $d_2 = 15$ (black) and $d_2 = 20$ (green). We also plot the entropy values for `Solo` case of traffic $\mathcal{T}_1$ in purple dashed line. When the ratios λ'_1:λ'_2 is large, meaning that the majority of the messages in the network are from type $\mathcal{T}_1$, the entropy values are almost the same for all values of d_2. However when this ratio λ'_1:λ'_2 declines, meaning that the majority of the message generation in the network are coming from $\mathcal{T}_2$ traffic we see that the best entropy values for target messages from $\mathcal{T}_1$ are when $d_2 = 5$ (blue lines). Having the slow traffic with large per-mix delays compared to the fast traffic does not provide the best anonymity for the fast traffic, however blending these two traffic together does still provide better anonymity than the `Solo` case (purple dashed line). As for the traffic $\mathcal{T}_2$ in 4b, the entropy values are to be expected: when we increase the per-mix delay d_2 we increase the anonymity for the messages from that traffic: the best entropy values are provided when $d_2 = 20$ (the green solid lines). We can summarize the insights as follows:

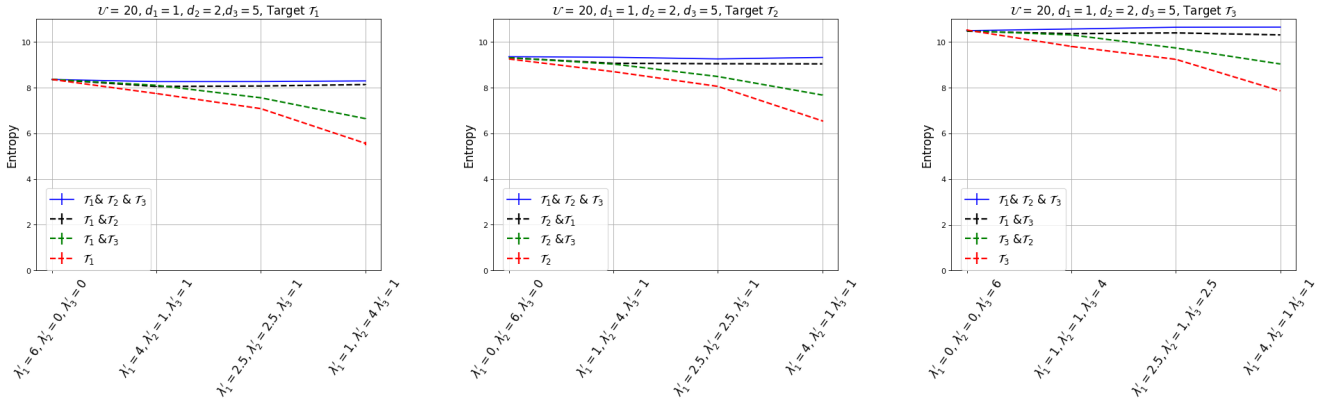
(1) When the amount of messages from the *fast* traffic is the majority in the network, the delay parameter of the *slow* traffic has less impact on the anonymity of the *fast* traffic.
(2) However, when the amount of the fast traffic is not the majority, the delay parameter of the slow traffic has a negative impact the anonymity of the fast traffic. As the difference between the delay parameters increases, anonymity slowly deteriorates.
(3) The rate delays of one traffic type impact not only the anonymity of messages within that specific type but also those from other types of traffic.

Remark. We consider in our analysis that the adversary observes the mixnet since the first message being sent, and therefore can count how many messages came to the mixnet, how many has left, and how many are still remaining. We argue, however, that changing the starting time of observation will not radically change our results since an adversary observing the network for a long time can infer these information with high confidence.

4.6 More Than Two Types of Traffic

To validate our insights obtained with two types of traffic, we evaluate the impact of blending three types of traffic with a set of small-scale experiments. We vary for different experiments the generation rates of each traffic such that the sum of the three rates of traffic generations $(\lambda'_1 + \lambda'_2 + \lambda'_3)$ is equal to 6 with $\mathcal{U} = 20$, meaning that we generate 120 messages per second, and the total number of messages is equal to 12000. Note that the number of clients as well as the number of messages does not change our insights regarding the blending strategy. In order to compute the probability of an output message being the target input message, given that the adversary knows the type of traffic of the target message when there are three types of traffic, we need to consider all possible values of k_2, which is the number of messages of type $\mathcal{T}_2$ that entered the node. We slightly update Algorithm 1 to compute the probability distribution over all possible k_2 values based on the derivations in Section 3.5, and present the updated algorithm in Appendix B.

We evaluate the impact of blending three types of traffic in a network with one mixnode: traffic type $\mathcal{T}_1$ with an average delay $d_1 = 1$, traffic type $\mathcal{T}_2$ with an average delay $d_2 = 2$, and traffic type



(a) Entropy for target messages of type $\mathcal{T}_1$ **(b) Entropy for target messages of type $\mathcal{T}_2$** **(c) Entropy for target messages of type $\mathcal{T}_3$.**

Figure 5: Evaluation of anonymity in terms of entropy for continuous mixnets with $L = 1$, $W = 1$, average delay $d_1 = \frac{1}{\lambda_1} = 1$ for traffic type $\mathcal{T}_1$, average delay $d_2 = \frac{1}{\lambda_2} = 2$ for traffic type $\mathcal{T}_2$, and average delay $d_3 = \frac{1}{\lambda_3} = 5$ for traffic type $\mathcal{T}_2$,

$\mathcal{T}_3$ with an average delay $d_3 = 5$. In Figure 5a, we present entropy values for target messages from traffic type $\mathcal{T}_1$ under different scenarios: when there are messages only from $\mathcal{T}_1$ (red), when there are messages from both $\mathcal{T}_1$ and $\mathcal{T}_2$ (black), when there are messages from $\mathcal{T}_1$ and $\mathcal{T}_3$ (green), and when there are messages from all three types of traffic (blue). Similarly, we repeat this analysis for target messages from $\mathcal{T}_2$ in Figure 5b and for $\mathcal{T}_3$ in Figure 5c.

In all three figures, there is a noticeable increase in entropy values when blending different types of traffic. Specifically, the highest entropy values are observed when messages from all three traffic types are combined.

5 DISCUSSION AND CONCLUSION

5.1 Broader Impact and Future Work

We have showed, through a diverse set of experiments in multiple settings, that the blending traffic types improves anonymity. The degree of this improvement, however, depends on the different average delays of the traffics and their respective generation ratios. In this paper, we assume that the end-to-end latency is set by the application using the mixnet-based system, leaving users with limited control over this aspect. However, envisioning a more user-centric mixnet-based system can consider the possibility of allowing users to choose delays to improve anonymity, as proposed by the authors in [16]. In this paper, the authors let senders specify for each message whether they prefer security or speed and hence end-to-end delays is chosen by the users. In such a scenario, a systematic study of the effects of different generation ratios along with the different average delays on anonymity is needed. Furthermore, we assumed in this paper that the adversary knows the type of traffic of input messages. However, it is not always straightforward for an adversary to deduce the types of messages, particularly when this information is not overtly leaked from client behaviors. An in-depth exploration of different applications that enable adversaries to infer message types is deferred for future research. In addition, future research should also consider more real-world scenarios: while our analysis and insights remain the same irrespective of message volume or the geo-location of the different mixes, exploring the impact of real data in practical settings may yield valuable additional insights. Such analysis will not reverse the relevance of our observations but may reveal nuanced insights that can be crucial in real-world implementations. Finally, due to limitations in simulation constraints, we regrettably had to work with a relatively small number of messages per second. Exploring the impact of blending different traffic types on anonymity in scenarios with a substantial volume of traffic presents an intriguing avenue for future research. Investigating by how much does the anonymity increases with blending traffic in high-volume situations could provide valuable insights for a more comprehensive understanding of the blending traffics.

5.2 Conclusion

We have provided the first quantitative analysis of the anonymity offered by continuous mixnets when multiple different traffic types are blended together. To that end, we provided (i) a novel analytical framework to compute the probabilities connecting the input and output messages of a mixnode when different traffic types with different latency requirements are blended together; (ii) a simulation-based evaluation of anonymity based on the proposed analytical framework considering varying proportions of traffic types, different average delays per traffic type, and diverse network settings. Our evaluations reveal that blending different traffic types through a mixnet enhances anonymity compared to dedicating a different mixnet to each traffic type.

ACKNOWLEDGMENTS

This research is partially supported by the Research Council KU Leuven under the grant C24/18/049, by CyberSecurity Research Flanders with reference number VR20192203, and by DARPA FA8750-19-C-0502. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of any of the funding agencies.

REFERENCES

[1] Nikolaos Alexopoulos, Aggelos Kiayias, Riivo Talviste, and Thomas Zacharias. 2017. MCMix: Anonymous Messaging via Secure Multiparty Computation. In *26th USENIX Security Symposium, USENIX Security 2017*, Engin Kirda and Thomas Ristenpart (Eds.). USENIX Association, Vancouver, BC, Canada, 1217–1234. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/alexopoulos

[2] Michael Backes, Aniket Kate, Praveen Manoharan, Sebastian Meiser, and Esfandiar Mohammadi. 2013. AnoA: A Framework for Analyzing Anonymous Communication Protocols. In *2013 IEEE 26th Computer Security Foundations Symposium*. IEEE Computer Society, New Orleans, LA, USA, 163–178. https://doi.org/10.1109/CSF.2013.18

[3] Iness Ben Guirat and Claudia Diaz. 2022. Mixnet optimization methods. *Proceedings on Privacy Enhancing Technologies* 2022 (07 2022), 456–477. https://doi.org/10.56553/popets-2022-0081

[4] George Danezis. 2004. The Traffic Analysis of Continuous-Time Mixes. In *Privacy Enhancing Technologies, 4th International Workshop, PET (Lecture Notes in Computer Science, Vol. 3424)*, David M. Martin Jr. and Andrei Serjantov (Eds.). Springer, Toronto, Canada, 35–50. https://doi.org/10.1007/11423409_3

[5] George Danezis. 2005. The Traffic Analysis of Continuous-Time Mixes. In *Privacy Enhancing Technologies*, David Martin and Andrei Serjantov (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 35–50.

[6] George Danezis, Roger Dingledine, and Nick Mathewson. 2003. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *2003 IEEE Symposium on Security and Privacy (S&P 2003)*. IEEE Computer Society, Berkeley, CA, USA, 2–15. https://doi.org/10.1109/SECPRI.2003.1199323

[7] Debajyoti Das, Easwar Mangipudi, and Aniket Kate. 2022. OrgAn: Organizational Anonymity with Low Latency. *Proceedings on Privacy Enhancing Technologies* 2022 (07 2022), 582–605. https://doi.org/10.56553/popets-2022-0087

[8] Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. 2018. Anonymity Trilemma: Strong Anonymity, Low Bandwidth Overhead, Low Latency - Choose Two. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, San Francisco, California, USA, 108–126.

[9] Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. 2020. Comprehensive Anonymity Trilemma: User Coordination is not enough. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (2020), 356–383. https://doi.org/10.2478/popets-2020-0056

[10] Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. 2021. Divide and Funnel: a Scaling Technique for Mix-Networks. Cryptology ePrint Archive, Paper 2021/1685. https://eprint.iacr.org/2021/1685 https://eprint.iacr.org/2021/1685.

[11] Claudia Diaz. 2005. *Anonymity and Privacy in Electronic Services*. PhD dissertation. KU leuven. https://www.esat.kuleuven.be/cosic/publications/thesis-115.pdf

[12] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. 2021. The Nym Network. https://nymtech.net/nym-whitepaper.pdf. , 38 pages.

[13] Claudia Diaz, Steven J. Murdoch, and Carmela Troncoso. 2010. Impact of Network Topology on Anonymity and Overhead in Low-latency Anonymity Networks. In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies* (Berlin, Germany) *(PETS'10)*. Springer-Verlag, Berlin, Heidelberg, 184–201. http://dl.acm.org/citation.cfm?id=1881151.1881162

[14] Claudia Diaz and Bart Preneel. 2004. Taxonomy of Mixes and Dummy Traffic. In *Information Security Management, Education and Privacy (IFIP, Vol. 148)*. Springer, Toulouse, France, 215–230. https://doi.org/10.1007/1-4020-8145-6_18

[15] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. 2002. Towards Measuring Anonymity. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies (PET'02)*. Springer-Verlag, Berlin, Heidelberg, 54–68.

[16] Roger Dingledine, Andrei Serjantov, and Paul F. Syverson. 2006. Blending Different Latency Traffic with Alpha-mixing. In *Privacy Enhancing Technologies, 6th International Workshop, PET 2006, Cambridge, UK, June 28-30, 2006, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 4258)*, George Danezis and Philippe Golle (Eds.). Springer, Cambridge, UK, 245–257. https://doi.org/10.1007/11957454_14

[17] Iness Ben Guirat, Devashish Gosain, and Claudia Diaz. 2021. MiXiM: Mixnet Design Decisions and Empirical Evaluation. In *WPES '21: Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society, Virtual Event, Korea, 15 November 2021*. ACM, Virtual Event, Korea, 33–37. https://doi.org/10.1145/3463676.3485613

[18] Dogan Kesdogan, Jan Egner, and Roland Büschkes. 1998. Stop- and- Go-MIXes Providing Probabilistic Anonymity in an Open System. In *Information Hiding*, David Aucsmith (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 83–98.

[19] Christiane Kuhn, Martin Beck, Stefan Schiffner, Eduard Jorswieck, and Thorsten Strufe. 2019. On privacy notions in anonymous communication. *Proceedings on Privacy Enhancing Technologies* 2019, 2 (2019), 105–125.

[20] Albert Kwon, David Lazar, Srinivas Devadas, and Bryan Ford. 2016. Riffle. *Proceedings on Privacy Enhancing Technologies* 2016, 2 (2016), 115–134. https://people.csail.mit.edu/devadas/pubs/riffle.pdf

[21] Albert Kwon, David Lu, and Srinivas Devadas. 2020. XRD: Scalable Messaging System with Cryptographic Privacy. In *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, Santa Clara, CA, USA, February 25-27, 2020*, Ranjita Bhagwan and George Porter (Eds.). USENIX Association, Santa Clara, CA, USA, 759–776. https://www.usenix.org/conference/nsdi20/presentation/kwon

[22] Ania M. Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. 2017. The Loopix Anonymity System. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, Engin Kirda and Thomas Ristenpart (Eds.). USENIX Association, Vancouver, BC, Canada, 1199–1216. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/piotrowska

[23] Andrei Serjantov and George Danezis. 2002. 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 (Lecture Notes in Computer Science, Vol. 2482)*, Roger Dingledine and Paul F. Syverson (Eds.). Springer, San Francisco, CA, USA, 41–53. https://doi.org/10.1007/3-540-36467-6_4

[24] Jelle van den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. 2015. Vuvuzela: scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles, SOSP 2015, Monterey, CA, USA, October 4-7, 2015*, Ethan L. Miller and Steven Hand (Eds.). ACM, Monterey, CA, USA, 137–152. https://doi.org/10.1145/2815400.2815417

A ANALYSIS STRATEGY WHEN THE ADVERSARY DOES NOT KNOW THE TYPES OF INPUT MESSAGES TO THE MIXNET

It is possible that the adversary does not know the types of the incoming messages to the mixnet (possibly every client is generating both types of traffic). It is still possible to extend our analysis strategy in such scenarios.

Suppose, the total number of observed packets is $\mathcal{N}$, then we have $X_1 + X_2 = \mathcal{N}$ where $X_1 \sim \text{Poisson}(\lambda_1')$ and $X_2 \sim \text{Poisson}(\lambda_2')$. Then we can say for an incoming message m,

$$\Pr[m \in \mathcal{T}_1] = \sum_{i=0}^{\mathcal{N}} \frac{i}{\mathcal{N}} \times \Pr[X_1 = i | X_1 + X_2 = \mathcal{N}]$$

where $(X_1 | X_1 + X_2 = \mathcal{N}) \sim \text{Binom}(\mathcal{N}, \frac{\lambda_1'}{\lambda_1' + \lambda_2'})$. Therefore, the above quantity can be further reduced to:

$$\begin{aligned}\Pr[m \in \mathcal{T}_1] &= \frac{1}{\mathcal{N}} \sum_{i=0}^{\mathcal{N}} i \times \Pr[X_1 = i | X_1 + X_2 = \mathcal{N}] \\ &= \frac{1}{\mathcal{N}} \times \mathbf{E}\left[\text{Binom}\left(\mathcal{N}, \frac{\lambda_1'}{\lambda_1' + \lambda_2'}\right)\right] = \frac{\lambda_1'}{\lambda_1' + \lambda_2'}\end{aligned}$$

Therefore, the value of $\dfrac{\Pr[m \in \mathcal{T}_1]}{\Pr[m \in \mathcal{T}_2]}$ does not depend on the total number of messages passing through the mixnode, or the number of messages the adversary observes.

Now the overall calculation of mapping probabilities between the incoming and outgoing messages of a mixnode is similar to the previous subsection, where the adversary knows that an incoming message i belongs to a specific type with some probability $0 \le p_i \le 1$. However, the adversary does not know the type of the target message. Suppose, the target message $m_{\texttt{target}}$ is of type $\mathcal{T}_1$ with probability p and of type $\mathcal{T}_2$ with probability $1 - p$. Then the analysis for the whole mixnet needs to be done twice: once assuming $m_{\texttt{target}} \in \mathcal{T}_1$, and then assuming $m_{\texttt{target}} \in \mathcal{T}_2$. Suppose the probabilities of an outgoing message m_i' being the target message are $p_{i,1}$ and $p_{i,2}$ in those two analyses. Then the final probability that $m_i' = m_{\texttt{target}}$ is calculated as $p \cdot p_{i,1} + (1 - p) \cdot p_{i,2}$.

B ALGORITHM FOR THREE TYPES OF TRAFFIC

We present a simplified version of the methodology presented in Section 3.5 to calculate the probabilities for a given mixnode in Algorithm 2, considering that the mixnode is in the first layer, i.e., the adversary has the knowledge about the types of all input messages. We use this methodology for our simulations in Section 4.6.

Algorithm 2: Probability computation on a single node with three types of traffic.

Result: Updated $\Pr[m_i = m_t | m_t \in \mathcal{T}_1]$.
Initialize:
G, Q, R : arbitrarily expandable Lists ;
$G.append(1)$; $Q.append(0)$;
$k = 0$; $i = 0$;
if *event(receive(m_i))* **then**
  $k++$;
  for $j \leftarrow 0$ **to** k **do**
    for $k2 \leftarrow 0$ **to** k **do**

$$\begin{aligned} Q[j,k2] &= Q[j-1,k2] \cdot \Pr[m_i \in \mathcal{T}_1] \\ &\quad + Q[j,k2-1] \cdot \Pr[m_i \in \mathcal{T}_2] \\ &\quad + Q[j,k2] \cdot \Pr[m_i \in \mathcal{T}_3] \\ &\quad + G[j-1] \cdot \Pr[m_i = m_t]; \end{aligned}$$

$$\begin{aligned} G[j,k2] &= G[j,k2-1] \cdot \Pr[m_i \in \mathcal{T}_2] \\ &\quad + G[j-1,k2] \cdot \Big(\Pr[m_i \in \mathcal{T}_1] - \Pr[m_i = m_t]\Big) \\ &\quad + G[j,k2] \cdot \Pr[m_i \in \mathcal{T}_3]; \end{aligned}$$

$$\begin{aligned} R[j,k2] &= R[j-1,k2] \cdot \Pr[m_i \in \mathcal{T}_1] \\ &\quad + R[j,k2-1] \cdot \Pr[m_i \in \mathcal{T}_2] \\ &\quad + R[j,k2] \cdot \Pr[m_i \in \mathcal{T}_3]; \end{aligned}$$

    end
  end
end
if *event(send(m_i))* **then**
  define $\text{denom}(j,k2) = j \cdot \lambda_1 + k2 \cdot \lambda_2 + (k - i - k2j) \cdot \lambda_3$;

$$\Pr[m_i = m_t] = \sum_{j=0}^{k-1} \sum_{k2=0}^{k-1} \frac{\lambda_1}{\text{denom}(j,k2)} \cdot Q[j,k2];$$

  for $j \leftarrow 0$ **to** k **do**
    for $k2 \leftarrow 0$ **to** k **do**

$$\begin{aligned} R[j,k2] &= \frac{\lambda_1}{\text{denom}(j+1,k2)} \cdot Q[j+1,k2] \\ &\quad + \frac{(j+1) \cdot \lambda_1}{\text{denom}(j+1,k2)} \cdot R[j+1,k2] \\ &\quad + \frac{(j,k2+1) \cdot \lambda_2}{\text{denom}(j,k2+1)} \cdot R[j,k2+1] \\ &\quad + \frac{(k-i-k2-j) \cdot \lambda_3}{\text{denom}(j,k2)} \cdot R[j]; \end{aligned}$$

$$\begin{aligned} Q[j,k2] &= \frac{j \cdot \lambda_1}{\text{denom}(j+1,k2)} \cdot Q[j+1] \\ &\quad + \frac{(k2+1) \cdot \lambda_2}{\text{denom}(j,k2+1)} \cdot Q[j,k2+1] \\ &\quad + \frac{(k-i-k2-j) \cdot \lambda_3}{\text{denom}(j,k2)} \cdot Q[j,k2]; \end{aligned}$$

$$\begin{aligned} G[j,k2] &= \frac{(j+1) \cdot \lambda_1}{\text{denom}(j+1,k2)} \cdot G[j+1,2] \\ &\quad + \frac{(k2+1) \cdot \lambda_2}{\text{denom}(j,k2+1)} \cdot G[j,k2+1] \\ &\quad + \frac{(k-i-k2-j) \cdot \lambda_3}{\text{denom}(j,k2)} \cdot G[j,k2]; \end{aligned}$$

    end
  end
  Forward Message (m_i);
  $i++$;
end