Are continuous stop-and-go mixnets provably secure?

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

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

Aggelos Kiayias
University of Edinburgh
IOG
United Kingdom
aggelos.kiayias@ed.ac.uk

Thomas Zacharias
University of Glasgow
United Kingdom
thomas.zacharias@glasgow.ac.uk

ABSTRACT

This work formally analyzes the anonymity guarantees of continuous stop-and-go mixnets and attempts to answer the titular question. Existing mixnet based anonymous communication protocols that aim to provide provable anonymity guarantees rely on round-based communication models, which requires synchronization among all the nodes and clients that is difficult to achieve in practice. Continuous stop-and-go mixnets (e.g., Loopix and Nym) provide a nice alternative by adding a random delay for each message on every hop independent of all other hops and all other messages. The core anonymization technique of continuous mixnets combined with the fact that the messages are sent by the clients to the mixnet at different times makes it a difficult problem to formally prove security for such mixnet protocols; existing end-to-end analyses for such designs provide only experimental evaluations for anonymity and were lacking a comprehensive formal treatment.

We are the first to close that gap and provide a formal analysis. We provide two indistinguishability based definitions (of sender anonymity), namely pairwise unlinkability and user unlinkability, tuned specifically for continuous stop-and-go mixnets. We derive the adversarial advantage as a function of the protocol parameters for the two definitions. We show that there is a fundamental lower bound on the adversarial advantage δ for pairwise unlinkability; however, strong user unlinkability (negligible adversarial advantage) can be achieved if the users message rate (λ_u) is proportional to message processing rate (λ) on the nodes.

KEYWORDS

anonymity, mixnets, provable security

1 INTRODUCTION

Anonymous communication (AC) protocols based on mixnets [1, 3, 4, 10, 19, 21, 23, 29] aim to provide anonymity by rerouting packets over several hops and adding delays on every hop of messages that allow the messages to mix with each other. All mixnets that attempt to provide provable anonymity guarantees do so by relying on some kind of round based communication model [1, 10, 19, 20, 29] — it is difficult to implement such round structure in practice when there are thousands of nodes and millions of clients in the system. Continuous stop-and-go mixnets (or simply, *continuous mixnets*) [11, 16, 26] avoid such round-based communication by adding a random delay (chosen from a predefined distribution) on every hop of each message, independent of all other hops of the message as well as independent of all other messages. In fact, continuous mixnets are already deployed on the internet by Nym [11] to support various applications like Telegram and crypto-currency transactions.

Although attractive as a system-design choice, it was not yet known if continuous mixnets can provide provable anonymity guarantees. There exist early analyses that evaluate only a single mixnode [5, 16]. Existing end-to-end analyses [13, 26] rely on experimental evaluations of entropy of messages [12] for specific settings and parameter choices in terms of number of users, topology, choice of delays etc. Such evaluations cannot provide a comprehensive understanding about how the anonymity guarantees will vary with the variation of those parameters/settings. Our work attempts to solve that open problem by providing a formal analysis of the anonymity guarantees provided by such continuous mixnets.

One major challenge towards formally proving anonymity for continuous mixnets is that the users do not send their messages in batches, rather different messages arrive at the mixnet from clients at different times. Any anonymous communication protocol (even via a trusted third party) with bounded delay guarantees will inherently have some leakage in this setting.[1] We precisely quantify the above leakage, that we coin as 'FIFO attack' (first-in-first-out), with continuous mixing strategy in the presence of a global passive adversary even when all the nodes in the mixnet are honest (§4).

Based on the above insight, we consider two indistinguishability-based definitions of sender anonymity. The first one, called *user unlinkability*, corresponds to an adversary that observes all messages going through the network, but does not control the messages of the honest users, and attempts to track specific target messages. The second one, called *pairwise unlinkability*, allows a strong adversary that controls all the client messages except the challenge messages, and also controls when the challenge users initiate the

[1]For that reason, round-based protocols with provable guarantees require the clients to send messages in batches in order to avoid such leakage.

Proceedings on Privacy Enhancing Technologies 2024(4), 665–683

https://doi.org/10.56553/popets-2024-0136

challenge messages. Our definitions are improvements over existing indistinguishability-based definitions [17, 24], to more suitably capture the FIFO effect.

As the main highlight of this work, we derive the adversarial advantage δ as a function of protocol parameters of the mixnet in the presence of global passive adversaries that can additionally passively compromise some parties in the protocol based on the two definitions mentioned above (§6.2, §5.2). For our proofs, we consider generic and representative versions of continuous mixnets (§3) adopted from Loopix [26], but without its active attack resistance or other additional features. As corollaries, we derive the range of parameters for which provable strong anonymity (negligible adversarial advantage) is achieved. Our proofs and results provide useful insights:

(1) We identify a sufficient condition for two messages mixing with each other; this could be useful to prove anonymity guarantees for other variations of similar designs.

(2) We show that a single cascade mixnet design without compromised nodes achieves exactly the same level of anonymity as a trusted third party for the same delay parameters.

(3) When we consider *pairwise unlinkability*, increasing the number of hops provide diminishing returns for anonymity.

(4) The presence of compromised nodes and choice of multiple paths drastically degrades pairwise unlinkability.

(5) With *user unlinkability*, the protocol does not face the above problems and can provide strong anonymity (negligible adversarial advantage) if the client sends messages at a rate proportional to the rate parameter of the (exponential) delay distribution.

1.1 Related Works And Challenges

Even though continuous mixnets are around for more than two decades [16], it was not yet known if they can provide provable anonymity guarantees. Existing mixnet designs [1, 10, 19, 20, 22, 28, 29] that attempt to provide provable anonymity guarantees mainly rely on (1) batch processing, and (2) round based communication model. Because of the round based communication model, all the messages that arrive to an honest mixnode in a given round are shuffled by the mixnode and forwarded to the next mixnodes/destination. Therefore, two messages are shuffled with each other if they have met in an honest mixnode at least once. With batch processing, the protocol waits for all (or a threshold number of) users to send their messages, and then all those messages stay in the protocol for the same number of rounds — thus avoiding any leakage from end-to-end time correlations.

However, continuous mixnets introduce interesting challenges towards formally proving the anonymity guarantees since they do not implement any rounds or batches. Each user generates their own messages independent of all other users, and each message is delayed on a mixnode independent of all other messages. Therefore, there are no explicit shuffles (that happens in round-based models) among messages in continuous mixnet designs. Additionally, different messages arriving the mixnet at different times could leak significant information to the adversary (which we formalize as first-in-first-out or FIFO attack in Section 4.2).

Existing Analyses. Some early analyses [5, 16] on continuous mixnets focus on analyzing the mixing only on a single honest node. They provide some very useful insights: (1) they analyze the correlation between the incoming and outgoing messages of the single mixnode; (2) if the input messages are generated using Poisson distribution and the delays are sampled from exponential distribution, the mixnode acts as an M/M/∞ queue.

The first end-to-end analysis for continuous mixnets came in the form of Loopix [26]. They provide an empirical analysis based on experimental evaluations with a setup of 100 clients and a stratified topology of 3 layers and 3 nodes per layer. However, such an analysis only provides some evidence for the anonymity properties; and cannot answer questions like how that guarantee would scale for different numbers of users, different topology, different number of nodes per layer etc. Additionally, the specific probabilities also depend on the specific nodes that are compromised for the experimental instance. Our work provides the first formal treatment to continuous mixnets.

2 PROBLEM STATEMENT AND ROADMAP

2.1 System Model

In a mixnet-based AC protocol, we consider a set of clients $\mathcal{U}$ who act as senders of messages, and are denoted by $u_1, \ldots, u_N$. They make use of a set of *mixnodes* $\mathcal{I}$ that are responsible for routing the messages to finally deliver them to the intended recipients. Since our analysis focuses on the study of sender anonymity, we consider a single recipient party R. In the following paragraphs, we explain how this setting is instantiated in the continuous mixing paradigm.

Clients. In our system, each honest client acts independently of all other clients. Each client u_i generates traffic at a rate of λ_u following Poisson distribution.

Routing. We consider a source-routed mixnet based architecture [26] allowing clients to send messages anonymously using an overlay network of mixnodes, each sender of a message selects the route through the network until it reaches the receiver. Preparing a message for sending requires encrypting it with public key material of the mixnodes selected by the sender as intermediaries in the route. Upon receiving a message, mixnodes use their private keys to strip a layer of encryption and discover the next hop in the route. In source-routing, the client picks all the mixnodes for the path of a message, for a given path length k (where k is specified as a protocol parameter), independent of all other messages by the same client or other clients.

Continuous Mixing. Each message is delayed on every hop using exponential delays [15, 26] with parameter λ. The delay for every hop of a message is sampled typically, by the sender independent of all other hops and all other messages, and encoded in the Sphinx headers [6]. Upon receiving and decrypting a message, a mixnode extracts the delay from the header, holds it for that amount of time, and then forwards it to its next destination. Intuitively, such delays lead to a pool of messages within a mixnode, and the messages within the pool can be considered 'mixed' with each other. We do not consider any cover traffic from the users or the mixnodes for our proofs.

Adversary. We consider a probabilistic polynomial time (PPT) adversary that can observe (but not alter) all network traffic. The adversary can also perform passive and static corruptions of senders, the recipient R, and a subset of mixnodes. Passive and static corruption means that the adversary chooses the subset of corrupted parties before the protocol starts; the adversary then has access to the internal states of these c mixnodes, including all of their keys and random choices; however, the compromised parties still follow the protocol specifications.

We focus on provable anonymity guarantees against global passive adversaries and do not consider active attacks. How to model all possible active attacks (not only for continuous mixnets, but in general for anonymous communication) still remains an open problem. Additionally, we consider that cryptography is perfect, and we do not consider any fingerprinting attacks in our model.

2.2 Security Goals

In this work, we consider sender anonymity properties against global passive adversaries. Achieving sender anonymity also implies relationship anonymity for bidirectional communications [24]. We expect to see similar guarantees for recipient anonymity; however, the exact details are left for future work. We consider two versions of security definition for sender anonymity:

User Unlinkability. In our first definition, the adversary does not control the time when the challenge messages are released, and the content of any other messages from the honest users. This more closely captures the surveillance scenario where the adversary observes an interesting/disturbing message received by the recipient and then tries to figure out who among Alice and Bob could have sent that message. Informally, the protocol achieves anonymity according to this definition as long as a target message from Alice is 'mixed' with at least one message from Bob.

Pairwise Unlinkability. Our second definition is stronger; here, we consider that the adversary controls the time when the challenge messages are released to the challenge users, the content of all other messages from the honest users, and then tries to distinguish who among them have sent which of the challenge messages after they are received by the recipient. Such a definition is useful to capture stronger adversarial scenarios, e.g., in the context of whistleblowing where the adversary might release fake/tagged documents and observe the time of its release to identify the whistleblower.

2.3 Proof Technique And Interesting Results

As part of our proof technique, we quantify the leakage from different messages arriving at the mixnet at different times (which we formalize as FIFO attack in Section 4.2), and identify the explicit conditions for mixing despite such leakage, to derive the provable guarantees for continuous mixnets. Our proof technique consists of the following general steps:

(1) We identify a set of *sufficient* conditions (good event) which 'mixes' two messages on a mixnode, so that the adversary cannot tell except with negligible probability which of them was sent by which user even if the rest of mixnodes on the paths for both messages are compromised.

(2) Then, we compute the probability of such a good event for a specific hop of a given message.

(3) That allows us to compute the probability that no such good event occurred over the whole path of a given message — which directly translates to the maximum success probability of a global passive adversary.

Additionally, dealing with continuous random variables for delays has its own mathematical challenges: (1) the probability of two messages mixing/meeting on a hop is dependent on all previous hops; (2) traditional combinatorial techniques are not applicable anymore, and computing the conditional probabilities becomes significantly more difficult; (3) the convolutions of the random variables do not always have closed form expressions. We overcome those hurdles in our proofs to derive our bounds.

Sufficient Conditions for Mixing. When delays are sampled from exponential distribution, based on the memoryless property of the distribution, it can be shown that two honest messages are 'mixed' in the view of an adversary if they meet at an honest mixnode (the second message enters the mixnode before the first message departs), and they have the same number of hops remaining when they meet. If this happens, the two messages are mixed with each other even if the rest of the paths of both of the messages are completely compromised. We call this the *sufficient condition* for mixing. If the delays are sampled from a distribution which is not memoryless, these conditions are not sufficient for mixing anymore.

Quantifying FIFO Attack. We show that there is an inherent leakage from the different arrival (to the mixnet) time of the messages — with significant probability they preserve the same order as they entered. We show that, even against a trusted third party anonymizer, a global passive adversary has an inherent advantage when the delays are sampled from Erlang distribution $\mathsf{Erl}(k, \lambda)$ (equivalent delay of a k-hop mixnet). The result about our FIFO attack can be considered as an improvement over the generic impossibility results [8, 9] for AC protocols.

Results About User Unlinkability. We show that continuous mixnets can provide user unlinkability with adversarial advantage $\delta < \frac{1}{2} \cdot (1 - f \cdot (1 - \mathrm{c}))^k$ over random coin toss, where c is the fraction of compromised mixnodes in the system and f can be a constant if the users message rate (λ_u) is proportional to message processing rate (λ) on the nodes. For this proof we model the mixnet as a Jackson network [2] with each mixnode acting as an M/M/∞ queue, and derive the bounds assuming a steady state of the network.

Results About Pairwise Unlinkability. With pairwise unlinkability, we derive both the upper bound and lower bound on the adversarial advantage. We start with a single cascade mixnet with no compromised mixnodes, and show that it achieves the exact same level of pairwise unlinkability as a trusted third party anonymizer for the same end-to-end delay distribution. When the adversary can compromise some mixnodes in the mixnet, the quality of mixing degrades, and there can be a significant (non-negligible) additional leakage to the adversary compared to the FIFO leakage.

When there are many mixnodes to choose from for every hop of a message, we show that the chances for two messages meeting each other degrades drastically (compared to single cascade mixnets). The fundamental lower bound on the adversarial advantage δ converges with the leakage in FIFO attack for very high values of k (c.f. Theorem 5) even in the presence of compromised nodes. Besides, the upper bound on δ remains very high even for high values of k (c.f. Figure 5).

3 THE CONTINUOUS MIXING PARADIGM

3.1 Preliminaries

Exponential Distribution. The exponential distribution $\mathsf{Exp}(\lambda)$ with parameter $\lambda \in \mathbb{R}^+$ has probability density function

$$f_\lambda(x) := \lambda e^{-\lambda x}, \quad \text{where } x \geq 0 ,$$

and cumulative distribution function $F_\lambda(x) = 1 - e^{-\lambda x}$. The mean of a random variable X following $\mathsf{Exp}_\lambda(x)$ is $1/\lambda$. In addition, X satisfies the *memoryless property*:

$$\Pr[X > x + t \mid X > t] = \Pr[X > x] = e^{-\lambda x} .$$

Erlang Distribution. The Erlang distribution $\mathsf{Erl}(k, \lambda)$ with parameters $k \in \mathbb{Z}^+$ and $\lambda \in \mathbb{R}^+$ can be seen as the sum of k independent random variables following $\mathsf{Exp}(\lambda)$. We recall that $\mathsf{Erl}(k, \lambda)$ has probability density function

$$f_{k,\lambda}(x) := \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!}, \quad \text{where } x \geq 0 , \tag{1}$$

and cumulative distribution function

$$F_{k,\lambda}(x) := 1 - \sum_{n=0}^{k-1} \frac{(\lambda x)^n}{n!} e^{-\lambda x} . \tag{2}$$

We observe that $\mathsf{Exp}(\lambda)$ matches the Erlang $\mathsf{Erl}(1, \lambda)$. For the security analysis of our protocols, we will apply the following useful equalities.

Equality 1. *For every $k \in \mathbb{Z}^+$ and $\lambda \in \mathbb{R}^+$, it holds that*

$$\int_0^\infty \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} dx = 1 .$$

The above equality follows directly from the definition of the Erlang distribution $\mathsf{Erl}(k, \lambda)$. For the following equalities, the proofs are in Appendices A.1 and A.2.

Equality 2. *For every $n \in \mathbb{N}$ and $k \in \mathbb{Z}^+$, it holds that*

$$\sum_{j=0}^{n} \binom{k+j-1}{j} = \binom{n+k}{n} .$$

Equality 3. *For every $k \in \mathbb{N}$, it holds that* $\sum_{n=0}^{k} \frac{\binom{n+k}{n}}{2^{k+n}} = 1$.

3.2 Model of Continuous Mixing Protocols

To explain our proofs easily, we consider two representative versions of continuous mixing protocols. Both protocols use exponential delay sampling and mainly differ in the mixnode path selection process. The first protocol represents a simple study case, called *cascade continuous mixing* protocol, where the path is fixed according to a cascade of k mixnodes. This construction is mostly of theoretical interest and allows us to explore the essence and strength of continuous mixing as an anonymization technique. The second protocol, called *multi-path continuous mixing* protocol, captures a full-fledged protocol in the realistic setting where multiple paths in the mixnet are used by different users depending on their own trusts and the overall scalability requirement of the protocol.

3.2.1 The cascade continuous mixing protocol. Let $\mathsf{CCM}^{k,\lambda,\lambda_u}$ denote the cascade continuous mixing protocol, where k is a positive integer and λ, λ_u are positive real values. The execution of $\mathsf{CCM}^{k,\lambda,\lambda_u}$ is carried as follows:

(1) Each message travels through a fixed cascade of k hops, denoted by $\mathsf{MX}_1 \to \cdots \to \mathsf{MX}_k$, before getting delivered to the recipient.

(2) The sender then onion encrypts the message (using Sphinx [6] packet structure) for the cascade (including the recipient), and sends it to the first of the mixnode in the cascade, MX_1, after some delay sampled from exponential distribution $\mathsf{Exp}(\lambda_u)$.

(3) Each mixnode delays the messages also following an exponential distribution $\mathsf{Exp}(\lambda)$.

Remark 1. Generating messages with intervals sampled from exponential distribution $\mathsf{Exp}(\lambda_u)$ yields a message rate following Poisson distribution with average rate λ_u.

Remark 2. The aggregate delay imposed by the k mixnodes follows the Erlang distribution $\mathsf{Erl}(k, \lambda)$.

3.2.2 The multi-path continuous mixing protocol. Let $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ denote the multi-path continuous mixing protocol, where k is a positive integer and λ, λ_u are positive real values. The execution of $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ is carried as follows:

1. Following the designs of Loopix [26] and Nym [11], we consider a stratified topology where mixnodes are arranged in a number of layers, such that mixnodes in layer i receive messages from mixnodes in layer $i-1$ and send messages to mixnodes in layer $i+1$. The path length of message routes is determined by the number of layers, and is denoted by k. Further, we consider that each layer has exactly K mixnodes.
2. The sender of the message picks a path of length k by picking one mixnode uniformly at random from each layer, independent of the choices of other users or other messages.
3. The sender samples k independent values $x_1, \ldots, x_k$ from $\mathsf{Exp}(\lambda)$. They then onion encrypt the message for the path (including the recipient), and embed the values in the onions header such that only i-th mixnode can see the x_i value. Then they send it to the first of the mixnodes in the path after a delay sampled from $\mathsf{Exp}(\lambda_u)$.
4. Each mixnode delays a message for the amount of time specified by x_i.

We want to highlight that, even though we consider such a stratified topology for our analysis, our results are also valid for *free-routing* where the users can choose a hop for a message from all the available mixnodes in the whole mixnet. That case can be considered as a special case of stratified topology where each layer contains the same set of node. We elaborate on this further in Section 6.2.4.

Remark 3. In $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$, given that the the packets are onion encrypted, a compromised mixnode only learns the previous and the next party on the path of a message.

3.3 Conditions for Mixing

Based on the description of $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in Subsections 3.2.1 and 3.2.2, respectively, we provide sufficient conditions for the mixing of two messages in our protocols. In particular, we show that if the following conditions are true (and they all have to be true) on a mixnode for two messages, then the adversary cannot distinguish if the messages went out in the same order as they came in or they are swapped:

(1) the two messages are honest messages,

(2) they meet at an honest mixnode (which means the second message enters the mixnode before the first message leaves),

(3) the two messages have the same number of hops remaining when they meet.

The justification behind the above set of conditions comes from two facts: (i) exponential distribution is memoryless, (ii) an honest mixnode does not reveal the mapping between the input and output messages unless the adversary deduce them from external information. Suppose, the first message enters the mixnode at time t_1 and the second message at time t_2. The first message leaves at time t'_1 and the second at time t'_2. There are three possible cases:

$t'_1 \leq t_2$: the first message leaves before the second message can arrive, and hence, they do not meet.

$t_2 < t'_1 < t'_2$: the second message arrives before the first message leaves, and hence they meet. However, the first message leaves before the second message — they preserve order.

$t'_1 \geq t'_2$: the first message leaves after the second message leaves — which means they are swapped.

In the first case, they do not meet and our conditions for mixing are not satisfied. Also, it is trivial in this case for the adversary to identify the mapping between the input and output messages. In the second and third case, our conditions for mixing are satisfied. The only thing that remains to argue is that those two cases are equally likely. That follows from the memorylessness of the exponential distribution. Given $t_2 < t'_1$, the probability that $t'_1 < t'_2$ is 0.5, since both the delays follow the same exponential distribution. Formally, we prove the following lemma (proof in A.3).

Lemma 1. *Let t_1, t_2, t'_1, t'_2 as in Subsection 3.3 and $\tau := t_2 - t_1 \geq 0$. Then, the following hold:*

(1) $\Pr[t'_1 \leq t_2] = 1 - e^{-\lambda\tau}$ (i.e., the probability that the two messages do not meet in the mixnode is $1 - e^{-\lambda\tau}$).

(2) $\Pr[t'_1 < t'_2 | t_2 < t'_1] = \frac{1}{2}$ (i.e., the probability that the first message leaves the mixnode first is 0.5, given the two messages meet).

4 TRUSTED THIRD PARTY ANONYMIZER

A trusted third party (TTP) anonymizer receives messages and shuffles them. Since we are analyzing continuous mixnets, our TTP will shuffle messages by adding random delays — whenever a message comes it adds a random delay to that message, and releases the message after that chosen delay. If there are sufficient number of messages received by the TTP regularly, then each message will mix with enough number of other messages. However, different messages arriving at different times might somewhat preserve the order when they leave. And that inherently provides linkability to any adversary who is observing the incoming and outgoing messages.

In our case, we want to prove mixing property for a continuous mixnet that delays messages on every node following an exponential distribution. Our TTP anonymizer mimics that by adding an overall delay of $\mathsf{Erl}(k, \lambda)$ plus constant network delays for each hop.

Event: `Receive`(message m, sender $u \in \mathcal{U}$)
Read the internal time as Cl.
Sample a value $d \leftarrow \mathsf{Erl}(k, \lambda) + k \cdot d_{net}$
▷ d_{net} denotes the constant network latency for each link.
▷ d includes the link-delay between the sender and the TTP, but not the link-delay between the TTP and the recipient.
Add the pair $(m, \mathsf{Cl} + d)$ in the priority queue Υ in an increasing order of $\mathsf{Cl} + d$.

Event: `TimeFired`($(m, \mathsf{Cl}) \in \Upsilon$)
▷ The delay-time has elapsed for a message.
Dequeue (m, Cl) from Υ.
Send m to R.

Figure 1: The trusted third party $\mathsf{TTP}^{k,\lambda}$ interacting with the senders in $\mathcal{U}$ and the recipient R, parameterized by k, λ.

Analyzing such a TTP anonymizer allows us to evaluate the leakage solely from the end-to-end delays, without considering other leakages that might be achieved by the adversary by observing the network links between mixnodes, or compromising some of the mixnodes (we deal with those leakages in Sections 5 and 6). Below, we provide the concrete instantiation of the TTP anonymizer.

4.1 A Trusted Third Party for Continuous Mixing

The trusted third party $\mathsf{TTP}^{k,\lambda}$ interacts with the senders in $\mathcal{U}$ and the recipient R, and is parameterized by the number of hops k it is mimicking and the delay parameter λ. The senders provide $\mathsf{TTP}^{k,\lambda}$ with their messages over a secure channel, so that no information about the message content is leaked to the adversary. $\mathsf{TTP}^{k,\lambda}$ acts as a central mixing node that delivers the messages to R after adding a delay sampled from the Erlang distribution $\mathsf{Erl}(k, \lambda)$ plus the constant network delays, as described in Figure 1.

Assuming that the central mixing node is honest, the power of the adversary is limited to an observer that monitors incoming and outgoing traffic. As this sets the minimum power for a global passive adversary, the security of $\mathsf{TTP}^{k,\lambda}$ serves as an optimistic bound of the security expected by a typical continuous mixing construction, such as $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ described in Subsections 3.2.1 and 3.2.2. Therefore, it is meaningful to explore the level of security that $\mathsf{TTP}^{k,\lambda}$ offers.

We define the protocol $\mathsf{TTP}^{k,\lambda,\lambda_u}$ as the one that naturally derives from the description of $\mathsf{TTP}^{k,\lambda}$ in Figure 1, when the delay from the sender to $\mathsf{TTP}^{k,\lambda}$ follows the exponential $\mathsf{Exp}(\lambda_u)$ distribution.

In the following subsection, we present an attack on $\mathsf{TTP}^{k,\lambda,\lambda_u}$. Intuitively, this sets a threshold on the pairwise unlinkability that $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ can promise, as it will be formally presented in Section 6.2.

4.2 The FIFO Attack

4.2.1 Setting. We consider a simplified setting with (i) two senders u_0, u_1, (ii) a single recipient R, and (iii) $\mathsf{TTP}^{k,\lambda}$ as described in Figure 1. The system state is as follows: each sender has a single message in her buffer and the queue is empty, i.e. there are no prior pending messages. The senders u_0, u_1 send their messages to the recipient R that receives messages m_0, m_1. The goal of the mix is to provide *sender anonymity* against an adversary that controls R and is a global observer, i.e. to hide whether communication occurs in (1) a "*direct*" manner: i.e., u_0, u_1 sent m_0, m_1 to R, respectively, or (2) a "*cross*" manner: i.e., u_0, u_1 sent m_1, m_0 to R, respectively.

In the above setting, the messages m_0, m_1 are delivered to the R with the following delays added: (i) the delay from the sender to $\mathsf{TTP}^{k,\lambda}$ follows the exponential $\mathsf{Exp}(\lambda_u)$ distribution, and (ii) the delay from $\mathsf{TTP}^{k,\lambda}$ till the recipient destination follows the Erlang $\mathsf{Erl}(k, \lambda)$ distribution.

4.2.2 Description of the FIFO Attack. We define the adversary $\mathcal{A}_{\text{fifo}}$ that begins observation at some given time when the messages m_0, m_1 are in the sender's queues and are about to be delivered. By the memoryless property of $\mathsf{Exp}(\lambda_u)$ and the description of the system state, we may assume that observation begins at time 0. Then, $\mathcal{A}_{\text{fifo}}$ executes the following steps:

(1) It waits until it records the following time values:
   (a) $t_{s,0}$: when u_0 sends her (encrypted) message to $\mathsf{TTP}^{k,\lambda}$;
   (b) $t_{s,1}$: when u_1 sends her (encrypted) message to $\mathsf{TTP}^{k,\lambda}$;
   (c) $t_{r,0}$: when message m_0 is forwarded to R by $\mathsf{TTP}^{k,\lambda}$;
   (d) $t_{r,1}$: when message m_1 is forwarded to R by $\mathsf{TTP}^{k,\lambda}$.
(2) Then, it decides as follows:
   - If $t_{s,0} < t_{s,1}$ and $t_{r,0} < t_{r,1}$, then it outputs 'direct'.
   - If $t_{s,0} < t_{s,1}$ and $t_{r,0} \geq t_{r,1}$, then it outputs 'cross'.
   - If $t_{s,0} \geq t_{s,1}$ and $t_{r,0} < t_{r,1}$, then it outputs 'cross'.
   - If $t_{s,0} \geq t_{s,1}$ and $t_{r,0} \geq t_{r,1}$, then it outputs 'direct'.

In a nutshell, $\mathcal{A}_{\text{fifo}}$ guesses based on the prediction that messages input earlier to the mixing node are more likely to be delivered earlier to the intended recipient. This adversarial strategy relies on the following interesting observation: the overall end-to-end network traffic observed by a global observer is *not* memoryless, as delays added by $\mathsf{TTP}^{k,\lambda}$ follow $\mathsf{Erl}(k, \lambda) + k \cdot d_{net}$. This distribution has a significant "FIFO" bias, as it is fully analyzed in the following subsection.

4.2.3 Analysis of the FIFO attack. Without loss of generality, assume that u_0, u_1 provide the messages m_0, m_1, respectively, in a "direct" manner to R (due to symmetry and independence, the "cross" case can be analysed similarly). We denote the following random variables:

(1) The delay x_0 until m_0 is sent to $\mathsf{TTP}^{k,\lambda}$ by u_0.
(2) The delay x_1 until m_1 is sent to $\mathsf{TTP}^{k,\lambda}$ by u_1.
(3) The delay $y_0' = y_0 + k \cdot d_{net}$ of $\mathsf{TTP}^{k,\lambda}$ until m_0 is forwarded to R, i.e., the time m_0 stays in the continuous mixnet.
(4) The delay $y_1' = y_1 + k \cdot d_{net}$ of $\mathsf{TTP}^{k,\lambda}$ until m_1 is forwarded to R, i.e., the time m_1 stays in the continuous mixnet.

Clearly, $x_0, x_1 \sim \mathsf{Exp}(\lambda_u)$ while $y_0, y_1 \sim \mathsf{Erl}(k, \lambda)$.

By the description in Section 4.2.2, we have that $t_{s,0}, t_{s,1}, t_{r,0}, t_{r,1}$ are the time values of $x_0, x_1, x_0 + y_0', x_1 + y_1'$, that $\mathcal{A}_{\text{fifo}}$ observes, in the direct case. Thus, $\mathcal{A}_{\text{fifo}}$ wins when either one of the following events happen:

$E_{0<1}$: $x_0 < x_1$ and $x_0 + y_0' < x_1 + y_1'$, or
$E_{0\geq 1}$: $x_0 \geq x_1$ and $x_0 + y_0' \geq x_1 + y_1'$.

The following theorem provides a concrete evaluation of the success probability of the FIFO attack.

Theorem 1. *Let $\lambda_u \geq \lambda$. The FIFO attack on $\mathsf{TTP}^{k,\lambda}$ described in Section 4.2.2 has success probability*

$$\phi_{\lambda,\lambda_u}(k) = \begin{cases} 1 - 2 \cdot \sum\limits_{n=0}^{k-1} \sum\limits_{j=0}^{n} \dfrac{(-1)^j \lambda_u}{(\lambda_u - \lambda)^{j+1}} & \\ \times \left(\sum\limits_{i=0}^{n-j} \dfrac{\lambda_u \lambda^{n-i} \binom{k+i-1}{k-1}}{(\lambda_u + \lambda)^{n-j-i+1} 2^{k+i}} - \dfrac{\lambda^j \binom{k+n-j-1}{k-1}}{2^{k+n-j+1}} \right), & \lambda_u > \lambda \\ \dfrac{1}{2} + \dfrac{\binom{2k}{k}}{2^{2k+1}}, & \lambda_u = \lambda \end{cases}$$

When $\lambda_u = \rho\lambda$ for a constant $\rho > 1$ we have the alternative expression

$$\phi_{\lambda,\lambda_u}(k) = 1 - 2 \cdot \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \rho}{(\rho - 1)^{j+1}} \times \left(\sum_{i=0}^{n-j} \frac{\rho \binom{k+i-1}{k-1}}{(\rho + 1)^{n-j-i+1} 2^{k+i}} - \frac{\binom{k+n-j-1}{k-1}}{2^{k+n-j+1}} \right).$$

We refer to Appendix A.4 for the detailed proof of Theorem 1. For notation simplicity, we will use $\phi(k)$ when λ, λ_u are implicit.

Analysis of the Sequence $\phi(k)$. In order to analyze $\phi(k)$ we plot the function in Fig. 2 for different values of ρ for a range of $k \in [1, 100]$. We observe in those plots that $\phi(k)$ decreases as k increases, for a given value of ρ. In our plots, $\phi(k)$ approaches close to 0.5 for large k and $\rho \geq 4$. With smaller ρ values (e.g., 1 and 2), $\phi(k)$ values are still > 0.51 for the range of of the plotted k values. However, they also show a trend to decline with k, and we can expect them to approach 0.5 as k becomes very large.

For each of the plots, $\phi(k)$ rapidly drops for the smaller values of k; then, with increased values of k, $\phi(k)$ does not drop that rapidly. This shows that increasing the number of hops provide diminishing returns in terms of the probability of two messages being swapped in $\mathsf{TTP}^{k,\lambda}$, and in continuous mixnets in general.

We can observe that even when $\rho = 64$, the success probability $\phi(k)$ for the adversary remains 0.500442 for $k = 100$. This means that the adversary still has over $\approx 2^{-11}$ advantage over a random guess. For $\rho = 64$ and $k = 20$, the success probability $\phi(k)$ is still more than 0.501. For $\rho = 1$, the success probability $\phi(k)$ remains above 0.525 even for $k = 100$. Thus, the question remains whether protocols with such continuous mixing strategy can still achieve meaningful anonymity guarantees; we formally investigate this in the later sections.

Case $\lambda_u < \lambda$. We observe in Fig. 2 that the success probability $\phi_{\lambda,\lambda_u}(k)$ for the adversary increases as ρ decreases. This indicates that $\phi_{\lambda,\lambda_u}(k)$ is strictly greater than $\phi_{\lambda,\lambda}(k) = \frac{1}{2} + \frac{\binom{2k}{k}}{2^{2k+1}}$ when $\lambda_u < \lambda$. Intuitively, if λ_u is smaller, $t_{s,0}$ and $t_{s,1}$ have high variances; and therefore, there is a high chance of them being far apart, which makes it more difficult for them to swap. Since the advantage of the adversary is already significant for $\lambda_u = \lambda$, we skip a formal derivation for the case $\lambda_u < \lambda$ and mainly focus on the case $\lambda_u \geq \lambda$ for the rest of the paper. However, as part of our proof in Appendix A.4 we also add a mathematical explanation about why this inequality holds (c.f. A.4.1).

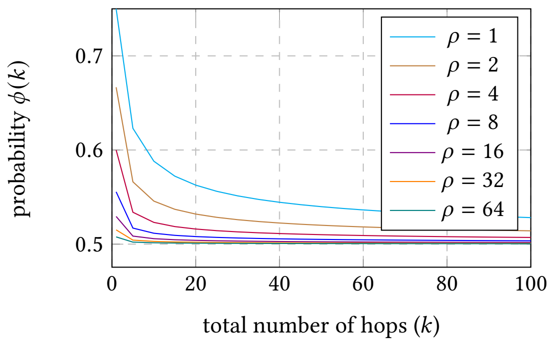


Figure 2: Success probability $\phi(k)$ of the FIFO attack over a k-hop Poisson mix when $\lambda_u = \rho\lambda$ for $k = 1, \ldots, 100$ and different values of ρ.

5 USER UNLINKABILITY PROPERTY

Here we study the anonymity of continuous mixnets in the context of our first security notion that we name *User Unlinkability*. Our formal treatment includes a game-based definition of the said notion and a rigorous assessment of the guarantees that multi-path continuous mixing provides.

5.1 User Unlinkability Definition

We assume an honest-but-curious global network level attacker that can eavesdrop on a fraction of the nodes (statically chosen), and has strong background knowledge about the behavior of the clients; formally, the attacker controls all but two users.

In user unlinkability, we formalize the question if a target message could have beeen swapped with a message from another user along the way. The adversary is not allowed to control the inception time for the target messages, and allows the honest users to choose the content of all other messages. We present our indistinguishability-based definition of user unlinkability via the corresponding game described in Fig. 3. In the *user unlinkability game*, the adversary does not control when the challenge message is generated, and only tries to backtrack the message after it is received by R. A message from Alice can be *mixed* with any of the messages sent by Bob. This property aims to capture the essence of real-world surveillance scenarios.

Definition 1 (User Unlinkability). *Let Π be a mixnet-based AC protocol with $\mathsf{N} > 2$ users and a set of mixing nodes $\mathcal{I}$. Let c be a non-negative number in $[0, 1)$. We say that Π provides* user unlinkability w.r.t. c with error $\delta(\cdot)$*, if it holds that*

$$\left| \max_{\mathcal{A} \in \mathrm{PPT}} \Pr\left[\mathcal{G}_{\mathsf{UL}}^{\Pi,\mathcal{A},\mathsf{c}}(1^\eta) = 1\right] - \frac{1}{2} \right| \leq \delta(\eta) \; .$$

5.2 Analysis for User Unlinkability

In order to analyse the user unlinkability guarantees, we first analyze some properties of the network flows in the mixnet. Based on those properties, we derive our bounds.

5.2.1 Estimates About Network Flows. In our case, the message generation is a Poisson process, and the processing on the mixnodes follows an exponential delay distribution. We prove our bounds by showing that the overall mixnet can be modeled as a Jackson network [2] with each node acting as an independent M/M/C queue: the arrivals are assumed to follow a Poisson process (Markovian or M), the service times are exponentially distributed (M), and C denotes the number of identical service channels or *servers*. We provide a brief introduction about queueing theory and notations in Appendix B.

The User Unlinkability game $\mathcal{G}_{\mathsf{UL}}^{\Pi,\mathcal{A},\mathsf{c}}(1^\eta)$.

- The challenger Ch provides the adversary $\mathcal{A}$ with the description of Π (that includes the description of the user set $\mathcal{U}$, the recipient R, and the mixing node set $\mathcal{I}$).
- $\mathcal{A}$ statically corrupts the recipient R, all users in $\mathcal{U}$ except from a pair of users u_0, u_1, and a subset of $\mathcal{I}$ denoted by $\mathcal{I}_{\mathsf{corr}}$. It provides Ch with (i) the description of $\mathcal{I}_{\mathsf{corr}}$; (ii) the identities of u_0, u_1.
- Ch generates the queues of messages for u_0 and u_1, those messages will be used for the protocol run.
- **Challenge:** before the start of the protocol run, $\mathcal{A}$ sends a challenge message m^* to Ch. In turn, Ch chooses a random bit $b \in \{0, 1\}$ and makes the following adjustments:
  - Pick a random position index x in the queue of u_b.
  - Add m^* to the queue of u_b at position x.

In any case, the recipient of all transmissions is R.
- Ch and $\mathcal{A}$ engage in an execution of Π where Ch first specifies the mixnet topology based on the protocol version (in our case, topology specification could be a random assignment of the cascade nodes in $\mathsf{CCM}^{k,\lambda,\lambda_u}$ or a random assignment of the nodes across the layers in $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$) and acts on behalf of u_0, u_1 and the mixing nodes in $\mathcal{I} \setminus \mathcal{I}_{\mathsf{corr}}$, while $\mathcal{A}$ controls the corrupted parties and monitors the network traffic as a global passive adversary.
- $\mathcal{A}$ can terminate the game any time by outputting a bit b^*.

The game returns a bit which is 1 if and only if the following conditions hold true:

C.1 $|\mathcal{I}_{\mathsf{corr}}| \leq \mathsf{c} \cdot |\mathcal{I}|$ (i.e., no more than c fraction of mixing nodes are corrupted).

C.2 $b^* = b$ (i.e., $\mathcal{A}$ guesses correctly).

Figure 3: The User Unlinkability game for protocol Π with N users against adversary $\mathcal{A}$ that corrupts up to a fraction of c mixing nodes.

Jackson Networks [2]. A network of H interconnected queueing nodes is a Jackson network if it has the following properties:

- external arrival to each queueing node i in the network follows a Poisson process with rate μ_i.
- All service times are exponentially distributed with parameter e_i and the service discipline is first-come, first-served (FCFS).
- A job leaving node i will either move to some new node j with probability $P_{i,j}$ or leave the network with probability q_i, where $q_i + \sum_{j=1}^{H} P_{i,j} = 1$.

If the above conditions are satisfied, according to Jackson's theorem [14], in a steady state of the network each node i can be considered as as independent M/M/C queue with arrival rate $\nu_i = \mu_i + \sum_{j=1}^{H} \nu_j \cdot P_{j,i}$; and the steady state condition is achieved if $\nu_i < e_i \cdot n_i$ for $i \in \{1, \ldots, H\}$ where n_i denotes the number of service channels (C) or *servers* in queueing node i. Intuitively, the steady state condition holds for Jackson networks if each queueing node has a sufficiently large number of *servers*. Later in the proof, we consider each mixnode can accept infinite number of messages (c.f. Remark 5), which makes $n_i \to \infty$ for all i, and the condition $\nu_i < e_i \cdot n_i$ trivially holds for all positive e_i values. The average number of jobs in the queue of node i in the steady state follows Poisson distribution with $\frac{\nu_i}{e_i}$.

Lemma 2. *For $k \geq 1$ and $\lambda_u, \lambda \in \mathbb{R}^+$, assuming constant delays on the network links, for the stream of messages sent by each client the cascade continuous mixnet $\mathsf{CCM}^{k,\lambda,\lambda_u}$ in the steady state has the followings properties:*

(1) each mixnode acts as an independent M/M/C queue with arrival rate λ_u;
(2) at any time the number of messages held by a mixnode follows Poisson distribution with average rate $\frac{\lambda_u}{\lambda}$.

Proof by construction. First we show that the cascade continuous mixnet $\mathsf{CCM}^{k,\lambda,\lambda_u}$ can be modeled as a Jackson network with k nodes. We consider the stream of messages from a single client u_{1-b}. We map the i-th mixnode on the cascade to the i-th node in the Jackson network. Each node i has the following properties:

(1) If $i = 1$, we have $\mu_i = \lambda_u$. Otherwise, $\mu_i = 0$.
(2) If each mixnode has a capacity to buffer up to C messages, the node i in the Jackson network can serve maximum C jobs in parallel, and each job takes time following exponential distribution with parameter $e_i = \lambda$.
(3) When a message leaves a node i, it goes to node $i + 1$ with probability $P_{i,i+1} = 1$ for $i < k$; and $P_{i,j} = 0$ for $j \neq i + 1$. The job exits the network with probability $q_k = 1$ for $i = k$, otherwise (when $i < k$) $q_i = 0$.

From the above observation, and the additional assumption that mixnodes process messages in FCFS manner, we can say that each mixnode in $\mathsf{CCM}^{k,\lambda,\lambda_u}$ acts as an M/M/C queue with arrival rate $\nu_i = \mu_i + \sum_{j=1}^{H} \nu_j P_{j,i} = \lambda_u$. From the properties of the Jackson network, we can also say that the number of messages in the queue of a node follows Poisson distribution with parameter $\dfrac{\nu_i}{e_i} = \dfrac{\lambda_u}{\lambda}$. □

Remark 4. In the above proof we assume that the network-link delays are constant. If the network-link delays are not constant, the mixnodes behave as ·/M/C queues instead of M/M/C queues. In that case, based on *Kleinrock independence approximation* [27], Lemma 2 is still a good approximation. We skip the detailed derivation of variable network-link delays or the exact accuracy of that approximation for future work.

Remark 5 (Assumption). If we consider that each mixnode has an infinite memory buffer, i.e., it can accept up to infinite number of messages, we have a special case of Jackson network where each node act as an M/M/∞ queue. In practice, a mixnode can have a system/memory limitation, and beyond that limit messages will be dropped. However, the number is generally high enough to avoid such message drops, and the approximation remains valid. In the following proofs, we consider that approximation and assume that each mixnode acts an an independent M/M/∞ queue in the steady state.

Lemma 3. *Let K, k be non-negative integers and $\lambda_u, \lambda \in \mathbb{R}^+$, assuming constant delays on the network links, and each mixnode has an infinite memory buffer, for the stream of messages sent by each client the multipath continuous mixnet $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in the steady state has the followings properties:*

(1) each mixnode acts as an independent M/M/∞ queue with arrival rate $\frac{\lambda_u}{\mathsf{K}}$;
(2) at any point of time the number of messages held by a mixnode follows Poisson distribution with rate parameter $\frac{\lambda_u}{\lambda \mathsf{K}}$.

Proof Sketch. The proof of this lemma is very similar to Lemma 2, except now each layer of the Jackson network has K nodes. Therefore, for a node i in layer h and another node j in layer $h+1$, $P_{i,j} = \frac{1}{\mathsf{K}}$ (assuming the node on each layer is chosen uniformly at random). And the rest of the proof follows Lemma 2. □

5.2.2 Anonymity Proof. With Lemma 3 at our disposal, we derive the user unlinkability guarantee provided by $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$. To prove user unlinkability, we first estimate the probability of at least one message from u_{1-b} present in a mixnode when the challenge message m^* arrives there. Then we compute the overall probability of m^* to meet at least one message from u_{1-b} on a path of length k.

Lemma 4. *For $k \geq 1$ and $\lambda_u, \lambda \in \mathbb{R}^+$, assuming constant delays on the network links, in a steady state of $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$, if a message m^* sent by u_b reaches i-th hop, the probability that there exists at least one message from user u_{1-b} also on i-th hop and on the same mixnode as m^* is given by,*

$$f = 1 - e^{-\frac{\lambda_u}{\lambda \cdot \mathsf{K}}}.$$

Proof. From Lemma 3 we know that the number of messages in a mixnode on hop i from each user follows Poisson distribution with parameter $\frac{\lambda_u}{\mathsf{K}\lambda}$. Therefore, when the message m^* reaches a mixnode on i-th hop, the probability that the mixnode holds at least one message from an arbitrary client u on the same i-th hop is given by,

$$\begin{aligned} f &= \Pr[X_u \geq 1] \qquad\qquad X_u \sim \mathsf{Poisson}\Big(\frac{\lambda_u}{\mathsf{K}\lambda}\Big) \\ &= 1 - e^{-\frac{\lambda_u}{\lambda \mathsf{K}}}. \end{aligned}$$

The same probability distribution holds for $X_{u_{1-b}}$ (denoting the number of messages from u_{1-b} held by a mixnode on i-th hop) for the client u_{1-b} as well. □

Intuitively, for any arbitrarily chosen moment, every mixnode in each layer holds at least one message from each client (including Alice, Bob, or any other arbitrarily chosen client Charlie) with high probability for appropriately chosen value of $\frac{\lambda_u}{\lambda \mathsf{K}}$. More specifically, if $\frac{\lambda_u}{\lambda \mathsf{K}}$ is a constant, the quantity f is also a constant; which means that the challenge message from Alice will encounter at least one message from Bob with significant probability, independent of the layer/hop i.

Theorem 2. *For $k \geq 1$ and $\lambda_u, \lambda \in \mathbb{R}^+$, assuming constant delays on the network links and a steady state of the network,* $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ *provides user unlinkability as defined in Definition 1 with error*

$$\delta \leq \frac{1}{2} \cdot (1 - f \cdot (1 - c))^k \text{, where } f = 1 - e^{-\frac{\lambda_u}{\lambda \mathsf{K}}}.$$

Proof. According to Lemma 4, the challenge message m^* on its i-th hop meets at least one message (also on i-th hop) from u_{1-b} with probability $f = 1 - e^{-\frac{\lambda_u}{\lambda \cdot \mathsf{K}}}$.

Since c fraction of mixnodes are compromised, and the mixnode on each hop is chosen uniformly at random, the probability that the i-th hop of m^* is honest is given by $(1 - c)$. Suppose, M'_i denotes the event that m^* does not mix with any message from Bob on its i-th hop. The probability that m^* does not mix with any message from u_{1-b} on any hops is given by,

$$\Pr\left[M'_1 \wedge \cdots \wedge M'_k\right] = \prod_{1 \leq i \leq k} \Pr[M'_i] = (1 - f(1 - \mathrm{c}))^k .$$

The above implies that

$$\begin{aligned}
&\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{UL}}^{\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] \\
&= 1 \cdot \Pr[M'_1 \wedge \cdots \wedge M'_k] + \frac{1}{2} \cdot \Pr[\neg\left(M'_1 \wedge \cdots \wedge M'_k\right)] \\
&= \frac{1}{2} + \frac{1}{2} \cdot \Pr[M'_1 \wedge \cdots \wedge M'_k] = \frac{1}{2} + \frac{1}{2}\left(1 - f(1 - \mathrm{c})\right)^k .
\end{aligned}$$

Therefore, $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ achieves user unlinkability with error $\delta \leq \frac{1}{2}\left(1 - f(1 - \mathrm{c})\right)^k$. □

Insights. We draw the following insights from Theorem 2:

(1) For constant f and c, $(1 - f(1 - c))$ is constant. So, the adversarial advantage δ declines rapidly with higher values of k.

(2) Consequently, for $k \in \omega(\log(\eta))$ we have an asymptotically negligible δ for the security parameter η.

(3) If $\frac{\lambda_u}{\lambda}$ is constant, f will go closer to 0 as K increases. To maintain the same level of δ, the number of hops k needs to grow with K. Typically, K increases with the number of users to support the increased number of users.

(4) k needs to grow approximately proportional to c to maintain the same level of δ, i.e., the increased fraction of compromised mixnodes can be compensated with increased end-to-end latency.

6 PAIRWISE UNLINKABILITY PROPERTY

In this section, we provide a formal study of the anonymity of continuous mixing, as captured by the description of $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ (cf. Subsections 3.2.1 and 3.2.2, respectively), under a stronger security notion that we name *Pairwise Unlinkability.* As in the case of user unlinkability, we begin by introducing a game-based definition of pairwise unlinkability. Subsequently, we investigate the level of anonymity that $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ can (or fail to) support.

6.1 Pairwise Unlinkability Definition

As in Subsection 5.1, we assume an honest-but-curious global network level attacker that can eavesdrop on a fraction of the nodes (statically chosen), and has strong background knowledge about the behavior of the clients; formally, the attacker controls all but two users.

The Pairwise Unlinkability game $\mathcal{G}_{\mathsf{PU}}^{\Pi,\mathcal{A},\mathrm{c}}(1^\eta)$.

- The challenger Ch provides the adversary $\mathcal{A}$ with the description of Π (that includes the description of the user set $\mathcal{U}$, the recipient R, and the mixing node set $\mathcal{I}$).
- $\mathcal{A}$ statically corrupts the recipient R, all users in $\mathcal{U}$ except from a pair of users u_0, u_1, and a subset of $\mathcal{I}$ denoted by $\mathcal{I}_{\mathsf{corr}}$. It provides Ch with (i) the description of $\mathcal{I}_{\mathsf{corr}}$; (ii) the identities of u_0, u_1.
- Ch and $\mathcal{A}$ engage in an execution of Π where Ch first specifies the mixnet topology based on the protocol version (in our case, topology specification could be a random assignment of the cascade nodes in $\mathsf{CCM}^{k,\lambda,\lambda_u}$ or a random assignment of the nodes across the layers in $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$) and acts on behalf of u_0, u_1 and the mixing nodes in $\mathcal{I} \setminus \mathcal{I}_{\mathsf{corr}}$, while $\mathcal{A}$ controls the corrupted parties and monitors the network traffic as a global passive adversary.
- **Challenge phase:** at any time, $\mathcal{A}$ sends a pair of challenge messages m_0, m_1 to Ch. In turn, Ch chooses a random bit $b \in \{0, 1\}$ and initiates two concurrent challenge transmissions according to the following cases:
  - If $b = 0$, then u_0 (resp. u_1) will begin the transmission of m_0 (resp. m_1).
  - If $b = 1$, then u_0 (resp. u_1) will begin the transmission of m_1 (resp. m_0).

  In any case, the recipient of both challenge transmissions is R.
- $\mathcal{A}$ can terminate the game any time by outputting a bit b^*.

The game returns a bit which is 1 if and only if the following conditions hold true:

C.1 $|\mathcal{I}_{\mathsf{corr}}| \leq \mathrm{c} \cdot |\mathcal{I}|$ (i.e., no more than c fraction of mixing nodes are corrupted).

C.2 $b^* = b$ (i.e., $\mathcal{A}$ guesses correctly).

Figure 4: The Pairwise Unlinkability game for protocol Π with N users against adversary $\mathcal{A}$ that corrupts up to a fraction of c mixing nodes.

In pairwise unlinkability, we formalize the question if the adversary could distinguish whether or not two messages, that travelled the same number of hops in the protocol, could have been swapped along the way. Pairwise unlinkability is close to message indistinguishability properties from the literature, such as tail indistinguishability by Kuhn et al. [17]. Capturing privacy-related notions via indistinguishability of two messages is commonly used by other provably secure designs [4, 7, 10, 19, 20]. We present our definition via the corresponding game described in Figure 4. In the *pairwise unlinkability game*, the adversary controls when the messages are initiated and observes when they are received by R. This reflects the background knowledge of the adversary about when a message of interest could have been generated, and the adversary can observe whose message (among Alice and Bob) enters first after that message has been generated. That helps us capture the essence of the FIFO attack that we detail in Section 4.2.

DEFINITION 2 (PAIRWISE UNLINKABILITY). *Let* Π *be a mixnet-based AC protocol with* $\mathsf{N} > 2$ *users and a set of mixing nodes,* $\mathcal{I}$. *Let* c *be a non-negative number in* $[0, 1)$. *We say that* Π *provides* pairwise unlinkability w.r.t. c with error $\delta(\cdot)$, *if it holds that*

$$\Big| \max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\Pi,\mathcal{A},\mathsf{c}}(1^\eta) = 1\right] - \frac{1}{2}\Big| \leq \delta(\eta) .$$

We say that a protocol achieves strong pairwise unlinkability *if* δ *is negligible in the security parameter* η.

6.2 Analysis for Pairwise Unlinkability

The definition of pairwise unlinkability is closely related to the FIFO attack presented in Section 4.2, except the adversary can now observe the (encrypted) messages after each intermediate hop, and some mixnodes might be corrupted. As we show in the next subsection, the success probability $\phi_{\lambda,\lambda_u}(k)$ in the FIFO attack against $\mathsf{TTP}^{k,\lambda}$ directly translates to the success probability $\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\Pi,\mathcal{A},\mathsf{c}}(1^\eta) = 1\right]$ against a k-hop cascade continuous mixnet $\mathsf{CCM}^{k,\lambda,\lambda_u}$ when there are no corrupted mixnodes (i.e., $\mathsf{c} = 0$). We extend our analysis for $\mathsf{CCM}^{k,\lambda,\lambda_u}$ with $\mathsf{c} > 0$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in the subsequent subsections.

6.2.1 Advantage of a Global Observers in $\mathsf{CCM}^{k,\lambda,\lambda_u}$ *Without Any Corrupted Nodes.* We prove that an adversary that acts as a global observer (but corrupts no mixing nodes) has no further advantage than a FIFO attacker, i.e., the FIFO attack is the best possible attack (in terms of pairwise unlinkability as defined in Definition 2) that can be launched in $\mathsf{CCM}^{k,\lambda,\lambda_u}$ when monitoring the network traffic. We prove that based on the following lemma.

LEMMA 5. *Let* m_x, m_y *be a pair of messages concurrently leaving from their senders to enter the same path in a* k*-hop continuous mix-net, with constant delays on the network links. Let* $x_0, \ldots, x_k$ *(resp.* $y_0, \ldots, y_k$*) be the delays added to* m_x *(resp.* m_y*) by the sender and the* k*-hops. Let* M *denote the event that* m_x *and* m_y *meet with each other in at least one of the hops. Then,* M *and* $\phi_{\lambda,\lambda_u}(k)$ *(as defined in Thm. 1) are related as follows:*

$$\frac{1}{2} + \frac{1}{2}\Pr\left[\neg M\right] = 1 - \frac{1}{2}\Pr\left[M\right] = \phi_{\lambda,\lambda_u}(k).$$

We present the detailed proof in Appendix A.5. Based on the above lemma, we can prove the following theorem about the anonymity guarantees of $\mathsf{CCM}^{k,\lambda,\lambda_u}$ when $\mathsf{c} = 0$.

THEOREM 3. *For every* $k \geq 1$, $\lambda_u, \lambda \in \mathbb{R}^+$, *it holds that*

$$\begin{aligned} &\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{CCM}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] \\ =&\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{TTP}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] = \phi_{\lambda,\lambda_u}(k). \end{aligned}$$

Therefore, the cascade continuous mix-net $\mathsf{CCM}^{k,\lambda,\lambda_u}$ *and the trusted third party anonymizer protocol* $\mathsf{TTP}^{k,\lambda,\lambda_u}$ *provide pairwise unlinkability w.r.t.* $\mathsf{c} = 0$ *with error* $\delta = \phi_{\lambda,\lambda_u}(k) - \frac{1}{2}$.

PROOF. Every attack against $\mathsf{TTP}^{k,\lambda,\lambda_u}$ can be directly translated to an attack against $\mathsf{CCM}^{k,\lambda,\lambda_u}$ with no mix-node corruptions (the attacker monitors the traffic at the end points of the communication). Therefore, we get the following inequality:

$$\begin{aligned} &\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{CCM}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] \\ \geq&\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{TTP}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right]. \end{aligned} \tag{3}$$

Since there are no corrupted mixnodes in our current consideration and the adversary against $\mathsf{CCM}^{k,\lambda,\lambda_u}$ only observes the encrypted messages entering and exiting the mixnodes for the intermediate hops, the probability of not satisfying the conditions for mixing (as specified in Section 3.3) is exactly the same as $\Pr[\neg M]$, where M denotes the event that the two messages meet with each other at least once. For $\mathsf{CCM}^{k,\lambda,\lambda_u}$ with $\mathsf{c} = 0$ we can say,

$$\begin{aligned} &\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{CCM}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] \\ =&1 \cdot \Pr[\neg M] + \frac{1}{2} \cdot \Pr[M] = \frac{1}{2} + \frac{1}{2} \cdot \Pr\left[\neg M\right]. \end{aligned} \tag{4}$$

Next, we model the FIFO attack presented in Section 4.2 in the context of the pairwise unlinkability game. In particular, we describe how $\mathcal{A}_{\mathsf{fifo}}$ engages in the game $\mathcal{G}_{\mathsf{PU}}^{\mathsf{TTP}^{k,\lambda,\lambda_u},\mathcal{A}_{\mathsf{fifo}},0}(1^\eta)$: when provided the user set $\mathcal{U}$, it sets (i) $\mathcal{I}_{\mathsf{corr}} = \emptyset$; (ii) a fixed pair (u_0, u_1) as the uncorrupted challenge senders (e.g., the first two identities in lexicographic order); (iii) the recipient R. At any time of its choice, it chooses a pair of distinct challenge messages m_0, m_1 and engages in the execution as described in Section 4.2.2. Instead of outputting 'direct' or 'cross', $\mathcal{A}_{\mathsf{fifo}}$ outputs 0 or 1, respectively.

By the above and Theorem 1, it is straightforward that

$$\Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{TTP}^{k,\lambda,\lambda_u},\mathcal{A}_{\mathsf{fifo}},0}(1^\eta) = 1\right] = \phi_{\lambda,\lambda_u}(k) . \tag{5}$$

From Lemma 5 we know that the probability $\Pr[\neg M]$ is related to $\phi_{\lambda,\lambda_u}(k)$. In turn, $\phi_{\lambda,\lambda_u}(k)$ directly translates to the success probability of $\mathcal{A}_{\mathsf{fifo}}$ in the FIFO game. So,

$$\begin{aligned} &\max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{CCM}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] \\ &= \frac{1}{2} + \frac{1}{2}\Pr\left[\neg M\right] = \phi_{\lambda,\lambda_u}(k) \qquad \triangleright \text{ Lemma 5} \\ &= \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{TTP}^{k,\lambda,\lambda_u},\mathcal{A}_{\mathsf{fifo}},0}(1^\eta) = 1\right] \qquad \triangleright \text{ Eq. (5)} \\ &\leq \max_{\mathcal{A} \in \mathsf{PPT}} \Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{TTP}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right]. \end{aligned} \tag{6}$$

By Eq. (3) and (6), the proof is complete. □

The following corollary of Lemma 5 simplifies the results for specific values of $k = 1, 2, 3$ which could be relevant to designs like Loopix [26] and Nym [11] where they consider $k = 3$.

COROLLARY 1. *Let* m_x, m_y *be a pair of messages concurrently leaving from their senders to enter the same path in a* k*-hop continuous mix-net with delay parameter* λ. *Let* $x_0, \ldots, x_k$ *(resp.* $y_0, \ldots, y_k$*) be the delays added to* m_x *(resp.* m_y*) by the sender and the* k*-hops. Let* M_j, $j = 1, \ldots, k$ *be the event that* m_x *and* m_y *meet at the* j*-th hop. Then, if we assume* $\lambda_u = \rho \cdot \lambda$, *where* $\rho \geq 1$, *for the first three layers, it holds that*

- $\Pr[M_1] = 1 - \frac{1}{1+\rho}$;
- $\Pr[M_1 \vee M_2] = 1 - \frac{\rho+2}{2(1+\rho)^2}$;
- $\Pr[M_1 \vee M_2 \vee M_3] = 1 - \frac{(3\rho+1)(\rho+4)}{8(1+\rho)^3}$.

6.2.2 Pairwise Unlinkability of $\mathsf{CCM}^{k,\lambda,\lambda_u}$ *Against Static Corruptions.* We analyze the level of anonymity that the cascade continuous mix-net provides against adversaries that (statically) corrupts a certain number of mixing nodes. Formally, we prove the following theorem.

Theorem 4. *Let k be non-negative integer,* $\mathsf{c} \in [0,1)$, $\lambda, \lambda_u \in \mathbb{R}^+$ *and* $\lambda_u \geq \lambda$. *The cascade continuous mix-net* $\mathsf{CCM}^{k,\lambda,\lambda_u}$ *provides pairwise unlinkability w.r.t.* c *with error* $\delta \leq \mathsf{c}(1-\phi(k)) + \phi(k) - \frac{1}{2}$.

Proof. Let us define the following two quantities:

- T is a random variable that denotes the total number of times the two challenge messages would meet in the protocol $\mathsf{CCM}^{k,\lambda,\lambda_u}$ based on the chosen delays. If $T = 0$, the two messages would not meet in $\mathsf{CCM}^{k,\lambda,\lambda_u}$, and the adversary definitely wins.
- $F(t)$ denotes the probability that t randomly chosen nodes are all compromised. Even if the two challenge messages meet t times in total, if those nodes are all compromised, the messages do not mix.

The actual value of $F(t)$ depends on how the k nodes in the cascade are chosen; however, we can say that $F(t+1) \leq F(t)$ since $0 \leq F(t) \leq 1$, and $F(1) = \mathsf{c}$.

Let us denote δ^* as the error for pairwise unlinkability provided by $\mathsf{CCM}^{k,\lambda,\lambda_u}$ when the adversary does not compromise any nodes. We know from Theorem 3 that $\delta^* = \frac{1}{2} \times \Pr[\neg M] = \phi(k) - \frac{1}{2}$, where M denotes the event that the two challenge messages meet on at least one node. For our current scenario, we can say the following about the event M' that the messages mix with each other :

$$\begin{aligned}
&\Pr[\neg M'] \\
=&\Pr[T=1]\cdot F(1) + \cdots + \Pr[T = \lfloor \mathsf{c}\cdot k \rfloor]\cdot F(\lfloor \mathsf{c}\cdot k \rfloor) + \Pr[T=0] \\
\leq&\Pr[T=1]\cdot F(1) + \cdots + \Pr[T = \lfloor \mathsf{c}\cdot k \rfloor]\cdot F(1) + \Pr[\neg M] \\
\leq&\Pr[T=1]\cdot F(1) + \cdots + \Pr[T = k]\cdot F(1) + \Pr[\neg M] \\
=&F(1)\times \Pr[M] + \Pr[\neg M] = \mathsf{c}\cdot 2\big(1-\phi(k)\big) + 2\big(\phi(k) - \frac{1}{2}\big).
\end{aligned} \tag{7}$$

From the above equation we can say,

$$\begin{aligned}
&\Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{CCM}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] = \frac{1}{2} + \frac{1}{2}\cdot\Pr[\neg M'] \\
\leq&\frac{1}{2} + \mathsf{c}\big(1-\phi(k)\big) + \big(\phi(k) - \frac{1}{2}\big) = \mathsf{c}(1-\phi(k)) + \phi(k).
\end{aligned}$$

□

The inequality step in Eq. (7) is untight and the error increases with large k values. However, for small values of c and small integers k our bound provides a reasonable upper bound on the adversarial advantage against the protocol.

6.2.3 Pairwise Unlinkability of $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$. Now we consider our multi-path continuous mixing protocol $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$: the formation of the message path is done via sampling one mixnode uniformly from each of the k layers. In the following theorem, we formally show the level of pairwise unlinkability expected in $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$.

Theorem 5. *Let* K, k *be non-negative integers,* $\lambda, \lambda_u \in \mathbb{R}^+$, $\lambda_u \geq \lambda$, *and* $\mathsf{c} \in [0,1)$. *The multipath continuous mixnet* $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ *provides pairwise unlinkability w.r.t.* c *with error* δ *where*

$$\left(1 - \frac{1-\mathsf{c}}{\mathsf{K}}\right)^k \cdot \big(1-\phi(k)\big) + \phi(k) \leq \delta + \frac{1}{2} \leq \left(1 - \frac{1-\mathsf{c}}{\mathsf{K}}\right) \cdot \big(1-\phi(k)\big) + \phi(k)$$

The proof of this theorem is very similar to that in Section 6.2.2, however the quantity $F(t)$ would be slightly different. With a single cascade, as long as the two messages have overlapping delays on a hop, they will meet. However, with many possible paths, meeting requires that the two messages also choose the same node on a given hop. This new factor in the proof captures this additional requirement, besides the necessity for the node being honest, for the two messages to meet. We include the detailed proof in Appendix A.6.

6.2.4 Free Routing. When the user picks the paths from all the available mixnodes in the mixnet, instead of following a stratified topology, the bounds remain the same if they choose the mixnodes on the path uniformly at random *with replacement.* The free routing topology with a total of K mixnodes can be considered as a special case of stratified topology where all the nodes are part of each layer. Since the user picks the nodes on the message path with replacement, all the probabilities in our bounds still hold. If the user picks a strategy to pick the mixnodes that is strictly better than selecting *with replacement*, the upper bound on adversarial advantage still holds.

Note that the same argument also holds for the bounds with user unlinkability in Section 5.2.

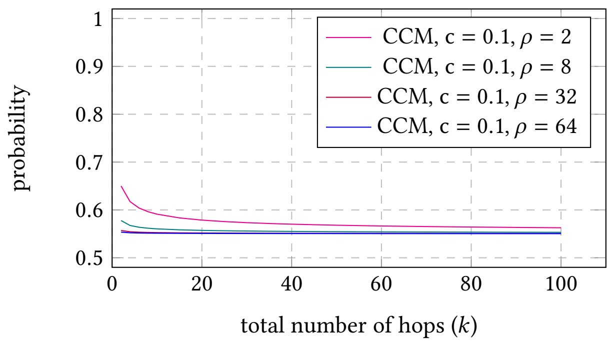
6.2.5 Analysis And Comparison With User Unlinkability. In Theorem 5, the upper bound on the error δ does not go to negligible for constant values of c and K, when $\mathsf{c} > 0$ or $\mathsf{K} > 1$. In Fig. 5, we plot the adversarial success probability for $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ with respect to the pairwise unlinkability game based on our proofs. For practical values of c and K the upper bound of the adversarial success probability remains significantly high (close to 1). Note that, for an overall adversarial success probability of 0.9 in the plot indicates 0.4 as an upper bound on δ. Moreover, the lower bound on the adversarial success probability is significant for practical values of total number of hops (for $\rho = 32, k = 15$, the adversarial success probability is 0.5198). For very high values of k, the lower bound comes close to 0.5, however, the untightness in our lower bound also plays a factor there.

We also plot in Fig. 5d the adversarial success probability with respect to the user unlinkability game, and the probability drops rapidly even for small values of ρ. Which provides strong confidence for the protocol when user unlinkability notion is used as the anonymity metric.

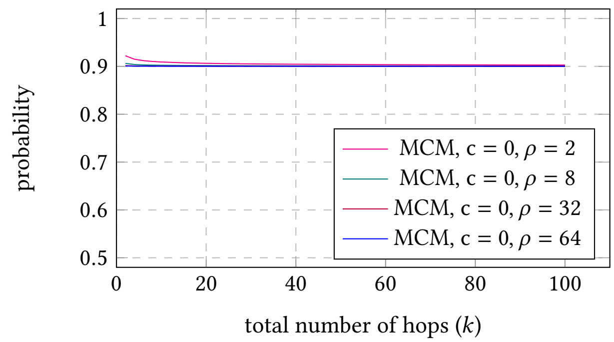
7 DISCUSSION AND CONCLUSION

7.1 About Round-based Protocols

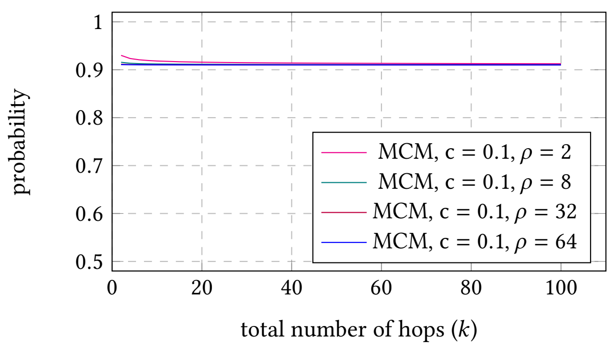
Round-based protocols [19, 21, 23, 29] assume some kind of batching or threshold model (where all the users send messages before the protocol starts a batch, or the protocol waits for a threshold number of messages) to achieve their provable security guarantees. There are no formal analyses about anonymity guarantees when the clients are allowed to send their messages in different rounds in a continuous manner, except the generic *impossibility* bounds [8, 9]. However, those generic bounds already tell us that such a leakage is



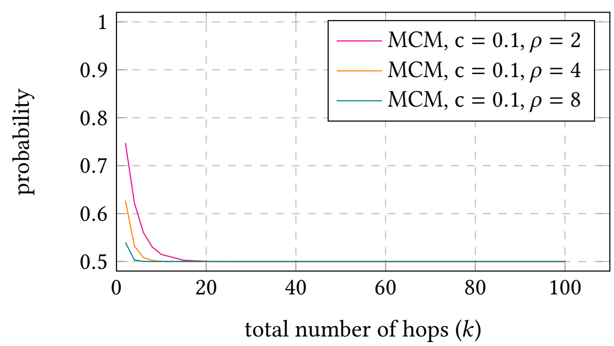
(a) Upper bound on the success probability (Thm. 4) of the adversary against $\mathsf{CCM}^{k,\lambda,\lambda_u}$ in the pairwise unlinkability game for $k = 1, \ldots, 100$, $\mathsf{c} = 0.1$, and different values of ρ.



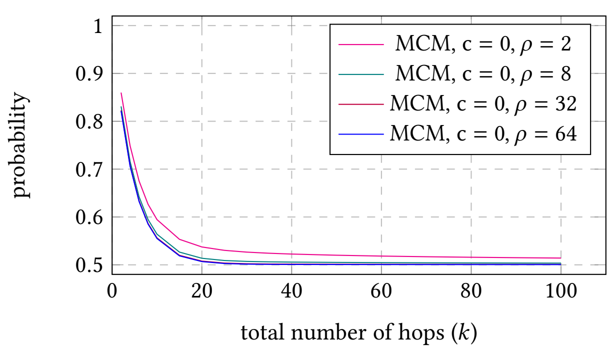
(b) Upper bound on the success probability (Thm. 5) of the adversary against $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in the pairwise unlinkability game for $k = 1, \ldots, 100$, $\mathsf{c} = 0$ (no compromised nodes), $\mathsf{K} = 5$ (no. of mixnodes per layer), and different values of ρ.



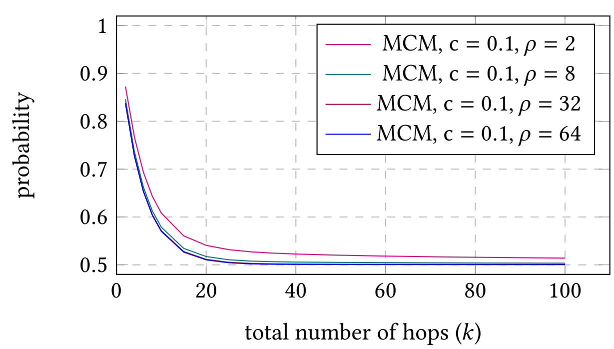
(c) Upper bound on the success probability (Thm. 5) of the adversary against $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in the pairwise unlinkability game for $k = 1, \ldots, 100$, $\mathsf{c} = 0.1$ (fraction of compromised nodes), $\mathsf{K} = 5$ (no. of mixnodes per layer), and different values of ρ.



(d) Upper bound on the success probability (Thm. 2) of the adversary against $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in the user unlinkability game for $k = 1, \ldots, 100$, $\mathsf{c} = 0.1$, $\mathsf{K} = 5$ (no. of mixnodes per layer), and different values of ρ.



(e) Lower bound on the success probability (Thm. 5) of the adversary against $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in the pairwise unlinkability game for $k = 1, \ldots, 100$, $\mathsf{c} = 0$ (no compromised nodes), $\mathsf{K} = 5$ (no. of mixnodes per layer), and different values of ρ.



(f) Lower bound on the success probability (Thm. 5) of the adversary against $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in the pairwise unlinkability game for $k = 1, \ldots, 100$, $\mathsf{c} = 0.1$ (fraction of compromised nodes), $\mathsf{K} = 5$ (no. of mixnodes per layer), and different values of ρ.

Figure 5: Analysis of the adversarial success probability of $\mathsf{CCM}^{k,\lambda,\lambda_u}$ and $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ in different settings.

fundamental to anonymous communication systems unless a high latency overhead is introduced, independent of the mixing strategy.

Although we have not derived the formal bounds, we conjecture that round-based mixnet designs will have a leakage similar to our analysis (c.f. Section 6.2) for pairwise unlinkability when the clients send their messages following a Poisson distribution and the delays (in number of rounds) are sampled from geometric distribution[2]. In such a setting, if messages stay on a node for only one round for each hop, the anonymity guarantees will be worse. A thorough analysis of such a setting for round-based mixnets is out of scope of this work and left for future work. Therefore, a verdict about which type of protocols (protocols with rounds or continuous mixnets) can provide better anonymity properties is not out yet.

7.2 Restrictions And Possible Extensions of the Definitions

Conceptually, pairwise unlinkability considers a really strong adversarial scenario where the adversary has the background information about when a specific message could have been generated. This would capture strong scenarios, e.g, whistleblowing, where the adversary might intentionally leak a fake/tagged document and observe the time of its release. Consequently, this notion is too strong to be achieved by the mixing strategy of continuous mixnets (as demonstrated by our pessimistic bounds in Section 6). Pairwise unlinkability also allows us to be comparable with the anonymity formalization for round-based mixnets (with batching). In Subsection 7.1 we discuss that such a timing leakage is inherent not only for continuous mixnets but for anonymous communication in general, when messages are received in a streaming manner.

Adding cover traffic might somewhat obfuscate that timing information, since the real message could be one among a series of messages sent by the sender. However, by extending the insight from our FIFO analysis, it could be conjectured that the adversary could gain a partial knowledge about which one of those sent packets could have been the actual message. That partial knowledge would also translate to the overall adversarial advantage. Derivation of the concrete bounds would require an improved proof technique:

(1) a tighter sufficient condition for mixing would have to consider that the cover messages from Alice mixing with cover messages (or the real message) from Bob could lead to overall mixing.

(2) since different packets could be the real message with potentially different probabilities (because of the FIFO effect), two packets meeting with each other in an honest node only adds to the probability of *mixing*, rather than satisfying the condition for mixing. The overall proof would require to cumulatively consider all such meets between all such messages.

On the other hand, user unlinkability is practically achievable without any cover traffic. However, it might fail to capture scenarios where a client sends a series of correlated messages with implicit ordering in their context, and the implicit ordering would potentially provide partial timing information to the adversary: the second message in the series has to be generated after the first message. Additionally, achieving user unlinkability guarantees requires the steady state assumption for the network, which is valid for most practical purposes. Nonetheless, the clients need to wait until the network reaches the steady state before they start sending useful messages.

Formally analyzing the effect of cover traffic on pairwise unlinkability and considering a meaningful relaxation (resp. tightening) of pairwise unlinkability (resp. user unlinkability) to capture realistic timing leakage are left for future works.

[2] since geometric distribution is a discrete approximation of exponential distribution.

7.3 Conclusion and Future Work

Our results provide a formal treatment for continuous mixnets for the first time and confirm strong guarantees for user unlinkability (Thm. 2). For pairwise unlinkability, we have a pessimistic upper bound and a lower bound (Thm. 5); and a tight lower bound for FIFO attack (Thm 3) on the success probability of the adversary. However, the treatment has room for improvements:

- We consider an adversary that compromises nodes before network orchestration. This implies that the adversary is not able to strategically corrupt. Still, this is a reasonable assumption for some mixnets e.g., Nym that randomize placement in regular (short) intervals. This also allows us to directly extend our results for free routing. Dealing with adaptive corruptions is an interesting direction for future work.
- Our results assume constant delays on the network links, which is not true in reality. However, we argue that network delays are clearly visible to global passive adversaries, and variable network delays does not change the insights significantly. A detailed mathematical derivation with variable network delays is left for future work.
- We did not consider any active attacks (packet drops, adding malicious delays for traffic pattern profiling) in our analysis. However, this is consistent with the analyses of existing provably secure mixnets, where they prove mixing guarantees against passive adversaries, and employ additional techniques (zero-knowledge proofs [18, 19, 28], trap messages [22, 25]) to detect and defend against such attacks. We consider a formal analysis of active attacks and corresponding defenses a complementary research problem.

Nevertheless, our analysis provides the first provable guarantees for continuous mixnets, and opens the path for further development in this direction.

ACKNOWLEDGMENTS

We thank Esfandiar Mohammadi, Sebastian Meiser, Harry Halpin, Alfredo Rial and Devashish Gosain for the insightful discussions. We thank the anonymous reviewers for their helpful comments. This research is partially supported by CyberSecurity Research Flanders with reference number VR20192203.

REFERENCES

[1] Nikolaos Alexopoulos, Aggelos Kiayias, Riivo Talviste, and Thomas Zacharias. 2017. MCMix: Anonymous Messaging via Secure Multiparty Computation. In *USENIX security symposium*. 1217–1234.
[2] Gunter Bolch, Stefan Greiner, Hermann Meer, and Kishor Trivedi. 2006. *Queuing Networks and Markov Chains—Modeling and Performance Evaluation with Computer Science Applications.* https://doi.org/10.1002/0471200581.ch6
[3] David Chaum. 1981. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Commun. ACM* 24, 2 (1981), 84–88. https://doi.org/10.1145/358549.358563
[4] David Chaum, Debajyoti Das, Farid Javani, Aniket Kate, Anna Krasnova, Joeri de Ruiter, and Alan T. Sherman. 2017. cMix: Mixing with Minimal Real-Time Asymmetric Cryptographic Operations. In *15th International Conference on Applied*

Cryptography and Network Security 2017.
[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 and Ian Goldberg. 2009. Sphinx: A Compact and Provably Secure Mix Format. 269–282.
[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] D. Das, S. Meiser, E. Mohammadi, and A. Kate. 2018. Anonymity Trilemma: Strong Anonymity, Low Bandwidth Overhead, Low Latency - Choose Two. In *2018 IEEE Symposium on Security and Privacy (SP)*. 108–126. https://doi.org/10.1109/SP.2018.00011 extended version under https://eprint.iacr.org/2017/954.
[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 (07 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.
[11] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. 2021. The Nym Network. https://nymtech.net/nym-whitepaper.pdf. , 38 pages.
[12] 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)*. 54–68.
[13] Iness 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
[14] James R Jackson. 1957. Networks of waiting lines. *Operations research* 5, 4 (1957), 518–521.
[15] 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.
[16] Dogan Kesdogan, Jan Egner, and Roland Büschkes. 1998. Stop- and- Go-MIXes Providing Probabilistic Anonymity in an Open System, Vol. 1525. 83–98. https://doi.org/10.1007/3-540-49380-8_7
[17] Christiane Kuhn, Martin Beck, and Thorsten Strufe. 2020. Breaking and (Partially) Fixing Provably Secure Onion Routing. In *Proceedings of the 41st IEEE Symposium on Security and Privacy*. IEEE, 168–185.
[18] Albert Kwon, Henry Corrigan-Gibbs, Srinivas Devadas, and Bryan Ford. 2017. Atom: Horizontally Scaling Strong Anonymity. In *Proceedings of the 26th Symposium on Operating Systems Principles* (Shanghai, China) *(SOSP '17)*. Association for Computing Machinery, New York, NY, USA, 406–422. https://doi.org/10.1145/3132747.3132755
[19] Albert Kwon, David Lu, and Srinivas Devadas. 2020. XRD: Scalable Messaging System with Cryptographic Privacy. In *Symposium on Networked Systems Design and Implementation.*
[20] Simon Langowski, Sacha Servan-Schreiber, and Srinivas Devadas. 2022. Trellis: Robust and Scalable Metadata-private Anonymous Broadcast. Cryptology ePrint Archive, Paper 2022/1548. https://eprint.iacr.org/2022/1548 https://eprint.iacr.org/2022/1548.
[21] David Lazar, Yossi Gilad, and Nickolai Zeldovich. 2018. Karaoke: Distributed private messaging immune to passive traffic analysis. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. 711–725.
[22] David Lazar, Yossi Gilad, and Nickolai Zeldovich. 2018. Karaoke: Distributed Private Messaging Immune to Passive Traffic Analysis. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. USENIX Association, Carlsbad, CA, 711–725. https://www.usenix.org/conference/osdi18/presentation/lazar
[23] David Lazar, Yossi Gilad, and Nickolai Zeldovich. 2019. Yodel: Strong Metadata Security for Real-Time Voice Calls. In *27thACM Symposium on Operating Systems Principles (SOSP 19)*.
[24] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi. 2016. AnoA: A Framework For Analyzing Anonymous Communication Protocols. *Journal of Privacy and Confidentiality (JPC)* 7(2), 5 (2016). Issue 2.
[25] Ania Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. 2017. The Loopix Anonymity System. In *Proc. 26th USENIX Security Symposium.*
[26] Ania M Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. 2017. The loopix anonymity system. In *26th USENIX Security Symposium (USENIX Security 17)*. 1199–1216.
[27] Adrian Popescu and Doru Constantinescu. 2011. *On Kleinrock's Independence Assumption*. Vol. 5233. 1–13. https://doi.org/10.1007/978-3-642-02742-0_1
[28] Nirvan Tyagi, Yossi Gilad, Derek Leung, Matei Zaharia, and Nickolai Zeldovich. 2017. Stadium: A Distributed Metadata-Private Messaging System. 423–440. https://doi.org/10.1145/3132747.3132783
[29] Jelle van den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. 2015. Vuvuzela: Scalable Private Messaging Resistant to Traffic Analysis. In *Proc. 25th ACM Symposium on Operating Systems Principles (SOSP 2015).*

A POSTPONED PROOFS

A.1 Proof of Equality 2

Proof. The proof is by induction on n for $n \geq 0$. Namely, for the base case $n = 0$ we have that for every $k \in \mathbb{Z}^+$:

$$\sum_{j=0}^{0} \binom{k+j-1}{j} = \binom{k-1}{0} = 1 = \binom{0+k}{0}.$$

Then, for the induction step, we have that

$$\sum_{j=0}^{n+1} \binom{k+j-1}{j} = \sum_{j=0}^{n} \binom{k+j-1}{j} + \binom{k+n+1-1}{n+1}$$
$$= \binom{n+k}{n} + \binom{n+k}{n+1} = \binom{n+1+k}{n+1}.$$

□

A.2 Proof of Equality 3

Proof. The equality is proven by the following observation: let z, w be two r.v.s that follow the $\mathsf{Erl}(k+1, \lambda)$ distribution independently. Like any pair of independent r.v.s that follow the same distribution, it holds that $Pr[z < w] = \Pr[z \geq w] = \frac{1}{2}$. If we compute the probability $Pr[z < w]$, then by Eq. (1) and (2), we get that

$$\begin{aligned}
&\frac{1}{2} = \Pr[z < w] \\
&= \int_0^\infty \frac{\lambda^{k+1} w^k e^{-\lambda w}}{k!} \int_0^w \frac{\lambda^{k+1} z^k e^{-\lambda z}}{k!} dz dw \\
&= \int_0^\infty \frac{\lambda^{k+1} w^k e^{-\lambda w}}{k!} \Big(1 - \sum_{n=0}^{k} \frac{(\lambda w)^n}{n!} e^{-\lambda w}\Big) dw \\
&= \int_0^\infty \frac{\lambda^{k+1} w^k e^{-\lambda w}}{k!} - \sum_{n=0}^{k} \int_0^\infty \frac{\lambda^{n+k+1} w^{n+k}}{n! k!} e^{-2\lambda w} dw \\
&= 1 - \sum_{n=0}^{k} \binom{n+k}{n} \int_0^\infty \frac{\lambda^{n+k+1} w^{n+k}}{(n+k)!} e^{-2\lambda w} dw \\
&= 1 - \sum_{n=0}^{k} \frac{\binom{n+k}{n}}{2^{n+k+1}} \int_0^\infty \frac{\lambda^{n+k+1} u^{n+k}}{(n+k)!} e^{-\lambda u} du \\
&= 1 - \sum_{n=0}^{k} \frac{\binom{n+k}{n}}{2^{n+k+1}}.
\end{aligned}$$

Thus, the equality follows from the above equality. □

A.3 Proof of Lemma 1

Proof. By the description of the continuous mixing, it holds that $t_1' = t_1 + \ell_1$, $t_2' = t_2 + \ell_2$ where $\ell_1, \ell_2 \sim \mathsf{Exp}(\lambda)$.

1. We have that

$$\Pr[t_1' \leq t_2] = \Pr[\ell_1 \leq \tau] = \int_0^\tau \lambda e^{-\lambda \ell_1} d\ell_1 = 1 - e^{-\lambda \tau}. \quad (8)$$

2. By the definition of conditional probability and Eq. (8),

$$\begin{aligned}\Pr[t_1' < t_2' | t_2 < t_1'] &= \frac{\Pr[t_1' < t_2' \wedge t_2 < t_1']}{\Pr[t_2 < t_1']} \\ &= \frac{\Pr[t_1' < t_2' \wedge t_2 < t_1']}{1 - \Pr[t_1' \leq t_2]} \\ &= e^{\lambda\tau} \cdot \Pr[t_1' < t_2' \wedge t_2 < t_1'].\end{aligned} \tag{9}$$

Next, by applying Equality 1, we compute

$$\begin{aligned}&\Pr[t_1' < t_2' \wedge t_2 < t_1'] = \\ =&\Pr[t_1 + \ell_1 < t_2 + \ell_2 \wedge t_2 < \ell_1 + t_1] \\ =&\Pr[\ell_1 < \ell_2 + \tau \wedge \ell_1 > \tau] \\ =&\int_0^\infty \lambda e^{-\lambda\ell_2} \int_\tau^{\ell_2+\tau} \lambda e^{-\lambda\ell_1} d\ell_1 d\ell_2 \\ =&\int_0^\infty \lambda e^{-\lambda\ell_2} \cdot \left(e^{-\lambda\tau} - e^{-\lambda(\ell_2+\tau)}\right) d\ell_2 \\ =&e^{-\lambda\tau} \cdot \left(\int_0^\infty \lambda e^{-\lambda\ell_2} d\ell_2 - \int_0^\infty \lambda e^{-2\lambda\ell_2} d\ell_2\right) \\ =&e^{-\lambda\tau} \cdot \left(1 - \frac{1}{2}\right) = \frac{e^{-\lambda\tau}}{2}.\end{aligned} \tag{10}$$

By Eq. (9) and (10), we get that

$$\Pr[t_1' < t_2' | t_2 < t_1'] = e^{\lambda\tau} \cdot \frac{e^{-\lambda\tau}}{2} = \frac{1}{2}. \qquad \square$$

A.4 Proof of Theorem 1

Proof. By the description of $\mathcal{A}_{\text{fifo}}$ we have that

$$\begin{aligned}&\Pr[\mathcal{A}_{\text{fifo}} \text{ wins}] = \phi_{\lambda,\lambda_u}(k) \\ &= \Pr[E_{0<1} \vee E_{0\geq 1}] = \Pr[E_{0<1}] + \Pr[E_{0\geq 1}] \\ &= \Pr[(x_0 < x_1) \wedge (x_0 + y_0' < x_1 + y_1')] + \\ &\quad + \Pr[(x_0 \geq x_1) \wedge (x_0 + y_0' \geq x_1 + y_1')] \\ &= \Pr[(x_0 < x_1) \wedge (x_0 + y_0 < x_1 + y_1)] + \\ &\quad + \Pr[(x_0 \geq x_1) \wedge (x_0 + y_0 \geq x_1 + y_1)] \\ &\qquad\qquad \triangleright\ y_i' = y_i + k \cdot d_{net}\ \forall i \in \{0, 1\}\end{aligned} \tag{11}$$

By the definition of $E_{0<1}$ and $E_{0\geq 1}$ and the symmetry of x_0, x_1 and $x_0 + y_0, x_1 + y_1$ we have that $\Pr[E_{0<1}] = \Pr[E_{0\geq 1}]$. So, it suffices that we compute the probability that event $E_{0<1}$ happens. We complete the proof in two parts: (1) when $\lambda_u > \lambda$, and (2) when $\lambda_u = \lambda$. In our analysis, we will apply the Equalities 1, 2, and 3.

Part 1: $\lambda_{\mathbf{u}} > \lambda$. We now proceed to the computation of $\Pr[E_{0<1}]$ when $\lambda_u > \lambda$. By the definition of x_0, x_1, y_0, y_1 and Eq. (1), we have that

$$\begin{aligned}&\Pr[(x_0 < x_1) \wedge (x_0 + y_0 < x_1 + y_1)] \\ =&\Pr[(x_0 < x_1) \wedge (y_0 < y_1 + x_1 - x_0)] \\ =&\int_0^\infty \frac{\lambda^k y_1^{k-1} e^{-\lambda y_1}}{(k-1)!} \int_0^\infty \lambda_u e^{-\lambda_u x_1} \int_0^{x_1} \lambda_u e^{-\lambda_u x_0} \\ &\cdot \int_0^{y_1+x_1-x_0} \frac{\lambda^k y_0^{k-1} e^{-\lambda y_0}}{(k-1)!} dy_0 dx_0 dx_1 dy_1\end{aligned} \tag{12}$$

We compute the probability in Eq. (12) by computing the following integrals:

By Eq. (2), we directly get that

$$\begin{aligned}A_1 &:= \int_0^{y_1+x_1-x_0} \frac{\lambda^k y_0^{k-1} e^{-\lambda y_0}}{(k-1)!} dy_0 \\ &= 1 - \sum_{n=0}^{k-1} \frac{\big(\lambda(y_1 + x_1 - x_0)\big)^n}{n!} e^{-\lambda(y_1+x_1-x_0)}.\end{aligned} \tag{13}$$

By Eq. (13), we get that

$$\begin{aligned}A_2 &:= \int_0^{x_1} \lambda_u e^{-\lambda_u x_0} A_1 dx_0 \\ &= \int_0^{x_1} \lambda_u e^{-\lambda_u x_0} dx_0 - \int_0^{x_1} \lambda_u e^{-\lambda_u x_0} \\ &\quad \cdot \sum_{n=0}^{k-1} \frac{\big(\lambda(y_1 + x_1 - x_0)\big)^n}{n!} \cdot e^{-\lambda(y_1+x_1-x_0)} dx_0 \\ &= \big[-e^{-\lambda_u x_0}\big]_0^{x_1} - e^{-\lambda_u(y_1+x_1)} \sum_{n=0}^{k-1} \frac{\lambda^n \lambda_u}{n!} \\ &\qquad \cdot \int_0^{x_1} (y_1 + x_1 - x_0)^n e^{(\lambda_u-\lambda)(y_1+x_1-x_0)} dx_0 \\ &= 1 - e^{-\lambda_u x_1} - e^{-\lambda_u(y_1+x_1)} \sum_{n=0}^{k-1} \frac{\lambda^n \lambda_u}{n!} \\ &\qquad \cdot \int_{y_1}^{y_1+x_1} z^n e^{(\lambda_u-\lambda)z} dz \qquad \triangleright\ z = y_1 + x_1 - x_0 \\ &= 1 - e^{-\lambda_u x_1} - e^{-\lambda_u(y_1+x_1)} \sum_{n=0}^{k-1} \frac{\lambda^n \lambda_u}{\cancel{n!}} \\ &\qquad \cdot \left[\sum_{j=0}^{n} \frac{(-1)^j \cancel{n!} z^{n-j} e^{(\lambda_u-\lambda)z}}{(n-j)!(\lambda_u - \lambda)^{j+1}}\right]_{y_1}^{y_1+x_1} \qquad \triangleright\ \lambda_u > \lambda \\ &= 1 - e^{-\lambda_u x_1} - e^{-\lambda_u(y_1+x_1)} \sum_{n=0}^{k-1} \lambda^n \lambda_u \\ &\qquad \cdot \left(\sum_{j=0}^{n} \frac{(-1)^j (y_1 + x_1)^{n-j} e^{(\lambda_u-\lambda)(y_1+x_1)}}{(n-j)!(\lambda_u - \lambda)^{j+1}}\right. \\ &\qquad\qquad \left. - \sum_{j=0}^{n} \frac{(-1)^j y_1^{n-j} e^{(\lambda_u-\lambda)y_1}}{(n-j)!(\lambda_u - \lambda)^{j+1}}\right) \\ &= 1 - e^{-\lambda_u x_1} - \sum_{n=0}^{k-1} \lambda^n \lambda_u \sum_{j=0}^{n} \frac{(-1)^j}{(n-j)!(\lambda_u - \lambda)^{j+1}} \\ &\qquad \cdot \left((y_1 + x_1)^{n-j} e^{-\lambda(y_1+x_1)} - y_1^{n-j} e^{-\lambda y_1 - \lambda_u x_1}\right)\end{aligned} \tag{14}$$

Subsequently, by Eq. (14), Eq. (2), and Equality 1, we get that

$$
\begin{aligned}
A_3 &:= \int_0^\infty \lambda_u e^{-\lambda_u x_1} A_2 dx_1 \\
&= \int_0^\infty \lambda_u e^{-\lambda_u x_1} dx_1 - \int_0^\infty \lambda_u e^{-2\lambda_u x_1} dx_1 \\
&\quad - \int_0^\infty \lambda_u e^{-\lambda_u x_1} \sum_{n=0}^{k-1} \lambda^n \lambda_u \sum_{j=0}^{n} \frac{(-1)^j}{(n-j)!(\lambda_u-\lambda)^{j+1}} \\
&\quad \cdot \left((y_1+x_1)^{n-j} e^{-\lambda(y_1+x_1)} - y_1^{n-j} e^{-\lambda y_1 - \lambda_u x_1}\right) dx_1 \\
&= 1 - \frac{1}{2} - \int_0^\infty \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda^n \lambda_u^2}{(n-j)!(\lambda_u-\lambda)^{j+1}} \\
&\quad \cdot \big((y_1+x_1)^{n-j} e^{-(\lambda_u+\lambda)(y_1+x_1)} e^{\lambda_u y_1} \\
&\qquad - y_1^{n-j} e^{-\lambda y_1} e^{-2\lambda_u x_1}\big) dx_1 \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda^n \lambda_u^2}{(\lambda_u-\lambda)^{j+1}} \\
&\quad \cdot \left(e^{\lambda_u y_1} \int_0^\infty \frac{(y_1+x_1)^{n-j}}{(n-j)!} e^{-(\lambda_u+\lambda)(y_1+x_1)} dx_1 \right. \\
&\qquad \left. - \frac{y_1^{n-j} e^{-\lambda y_1}}{(n-j)!} \int_0^\infty e^{-2\lambda_u x_1} dx_1 \right) \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda^n \lambda_u^2}{(\lambda_u-\lambda)^{j+1}} \left(\frac{e^{\lambda_u y_1}}{(\lambda_u+\lambda)^{n-j+1}} \right. \\
&\quad \left. \cdot \int_{y_1}^\infty \frac{\alpha^{n-j+1} z^{n-j}}{(n-j)!} e^{-\alpha z} dz - \frac{y_1^{n-j} e^{-\lambda y_1}}{2\lambda_u (n-j)!} \right) \\
&\qquad \triangleright\, z = y_1 + x_1, \alpha = \lambda_u + \lambda \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda^n \lambda_u^2}{(\lambda_u-\lambda)^{j+1}} \left(\frac{e^{\lambda_u y_1}}{(\lambda_u+\lambda)^{n-j+1}} \right. \\
&\quad \left. \cdot \sum_{i=0}^{n-j} \frac{(\lambda_u+\lambda)^i y_1^i}{i!} e^{-(\lambda_u+\lambda) y_1} - \frac{y_1^{n-j} e^{-\lambda y_1}}{2\lambda_u (n-j)!} \right) \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda^n \lambda_u^2}{(\lambda_u-\lambda)^{j+1}} \\
&\quad \left(\sum_{i=0}^{n-j} \frac{y_1^i e^{-\lambda y_1}}{i!(\lambda_u+\lambda)^{n-j-i+1}} - \frac{y_1^{n-j} e^{-\lambda y_1}}{2\lambda_u (n-j)!} \right)
\end{aligned}
\tag{15}
$$

Finally, by Eq. (12), (15) and applying the Equalities 2 and 3, we conclude that

$$
\begin{aligned}
&\Pr[(x_0 < x_1) \wedge (x_0 + y_0 < x_1 + y_1)] \\
&= \int_0^\infty \frac{\lambda^k y_1^{k-1} e^{-\lambda y_1}}{(k-1)!} A_3 dy_1 \\
&= \int_0^\infty \frac{\lambda^k y_1^{k-1} e^{-\lambda y_1}}{2\cdot(k-1)!} dy_1 - \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda^{n+k} \lambda_u^2}{(\lambda_u-\lambda)^{j+1}(k-1)!} \\
&\quad \int_0^\infty \left(\sum_{i=0}^{n-j} \frac{y_1^{k+i-1} e^{-2\lambda y_1}}{i!(\lambda_u+\lambda)^{n-j-i+1}} - \frac{y_1^{k+n-j-1} e^{-2\lambda y_1}}{2\lambda_u (n-j)!} \right) dy_1 \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda^{n+k} \lambda_u^2}{(\lambda_u-\lambda)^{j+1}(k-1)!} \left(\sum_{i=0}^{n-j} \frac{(k+i-1)!}{(2\lambda)^{k+i}} \right. \\
&\quad \left. \cdot \frac{1}{i!(\lambda_u+\lambda)^{n-j-i+1}} - \frac{(k+n-j-1)!}{2\lambda_u (n-j)!(2\lambda)^{k+n-j}} \right) \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda_u}{(\lambda_u-\lambda)^{j+1}} \\
&\quad \cdot \left(\sum_{i=0}^{n-j} \frac{\lambda_u \lambda^{n-i} \binom{k+i-1}{k-1}}{(\lambda_u+\lambda)^{n-j-i+1} 2^{k+i}} - \frac{\lambda^j \binom{k+n-j-1}{k-1}}{2^{k+n-j+1}} \right).
\end{aligned}
\tag{16}
$$

By Eq. (11) and (16), we conclude that

$$
\begin{aligned}
\phi_{\lambda,\lambda_u}(k) &= \Pr[\mathcal{A}_{\text{fifo}} \text{ wins}] = 2 \cdot \Pr[E_{0<1}] \\
&= 1 - 2 \cdot \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \lambda_u}{(\lambda_u-\lambda)^{j+1}} \\
&\quad \cdot \left(\sum_{i=0}^{n-j} \frac{\lambda_u \lambda^{n-i} \binom{k+i-1}{k-1}}{(\lambda_u+\lambda)^{n-j-i+1} 2^{k+i}} - \frac{\lambda^j \binom{k+n-j-1}{k-1}}{2^{k+n-j+1}} \right).
\end{aligned}
$$

By the above, when we consider $\lambda_u = \rho\lambda$ for a constant $\rho > 1$, it holds that

$$
\begin{aligned}
\phi_{\lambda,\lambda_u}(k) &= 1 - 2 \cdot \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \rho\lambda}{(\rho\lambda-\lambda)^{j+1}} \\
&\quad \cdot \left(\sum_{i=0}^{n-j} \frac{\rho\lambda \cdot \lambda^{n-i} \binom{k+i-1}{k-1}}{(\rho\lambda+\lambda)^{n-j-i+1} 2^{k+i}} - \frac{\lambda^j \binom{k+n-j-1}{k-1}}{2^{k+n-j+1}} \right) \\
&= 1 - 2 \cdot \sum_{n=0}^{k-1} \sum_{j=0}^{n} \frac{(-1)^j \rho}{(\rho-1)^{j+1}} \\
&\quad \cdot \left(\sum_{i=0}^{n-j} \frac{\rho \binom{k+i-1}{k-1}}{(\rho+1)^{n-j-i+1} 2^{k+i}} - \frac{\binom{k+n-j-1}{k-1}}{2^{k+n-j+1}} \right)
\end{aligned}
\tag{17}
$$

Part 2: $\lambda_\mathbf{u} = \lambda$. We compute $\Pr[E_{0<1}]$ as in Eq. (12) for the special case where $\lambda_u = \lambda$. We observe that A_1 remains unchanged, i.e., (13) still holds. Thus, by Eq. (13), we get

$$\begin{aligned}
A_2 &:= \int_0^{x_1} \lambda e^{-\lambda x_0} A_1 dx_0 \\
&= \left[-e^{-\lambda x_0}\right]_0^{x_1} - e^{-\lambda(y_1+x_1)} \sum_{n=0}^{k-1} \frac{\lambda^{n+1}}{n!} \\
&\quad \cdot \int_0^{x_1} (y_1 + x_1 - x_0)^n dx_0 \\
&= 1 - e^{-\lambda x_1} - \sum_{n=0}^{k-1} \frac{\lambda^{n+1}}{(n+1)!} \\
&\quad \cdot \left((y_1+x_1)^{n+1} - y_1^{n+1}\right) e^{-\lambda(y_1+x_1)} \ .
\end{aligned} \tag{18}$$

Subsequently, by Eq. (18), Eq. (2), and Equality 1, we get that

$$\begin{aligned}
A_3 &:= \int_0^{\infty} \lambda e^{-\lambda x_1} A_2 dx_1 \\
&= \int_0^{\infty} \lambda e^{-\lambda x_1} dx_1 - \frac{1}{2} \int_0^{\infty} 2\lambda e^{-2\lambda x_1} dx_1 \\
&\quad - \sum_{n=0}^{k-1} e^{\lambda y_1} \int_0^{\infty} \frac{\lambda^{n+2}}{(n+1)!} (y_1 + x_1)^{n+1} e^{-2\lambda(y_1+x_1)} dx_1 \\
&\quad + \sum_{n=0}^{k-1} \frac{\lambda^{n+1} y_1^{n+1}}{(n+1)!} e^{-\lambda y_1} \int_0^{\infty} \lambda e^{-2\lambda x_1} dx_1 \\
&= 1 - \frac{1}{2} - \sum_{n=0}^{k-1} \frac{e^{\lambda y_1}}{2} \int_{2y_1}^{\infty} \frac{\lambda^{n+2}}{(n+1)!} \left(\frac{z}{2}\right)^{n+1} e^{-\lambda z} dz \\
&\quad + \sum_{n=0}^{k-1} \frac{\lambda^{n+1} y_1^{n+1}}{2(n+1)!} e^{-\lambda y_1} \int_0^{\infty} 2\lambda e^{-2\lambda x_1} dx_1 \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{e^{\lambda y_1}}{2^{n+2}} \left(1 - F_{n+2,\lambda}(2y_1)\right) + \sum_{n=0}^{k-1} \frac{\lambda^{n+1} y_1^{n+1}}{2(n+1)!} e^{-\lambda y_1} \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{e^{\lambda y_1}}{2^{n+2}} \sum_{j=0}^{n+1} \frac{(2\lambda y_1)^j}{j!} e^{-2\lambda y_1} + \sum_{n=0}^{k-1} \frac{\lambda^{n+1} y_1^{n+1}}{2(n+1)!} e^{-\lambda y_1} \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{1}{2^{n+2}} \sum_{j=0}^{n+1} \frac{(2\lambda y_1)^j}{j!} e^{-\lambda y_1} + \sum_{n=0}^{k-1} \frac{\lambda^{n+1} y_1^{n+1}}{2(n+1)!} e^{-\lambda y_1} \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{1}{2^{n+2}} \sum_{j=0}^{n} \frac{(2\lambda y_1)^j}{j!} e^{-\lambda y_1} \ .
\end{aligned} \tag{19}$$

Finally, by Eq. (12), (19) and applying the Equalities 2 and 3, we conclude that

$$\begin{aligned}
&\Pr[E_{0<1}] = \Pr[(x_0 < x_1) \wedge (x_0 + y_0 < x_1 + y_1)] \\
&= \int_0^{\infty} \frac{\lambda^k y_1^{k-1} e^{-\lambda y_1}}{(k-1)!} A_3 dy_1 \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{1}{2^{n+2}} \sum_{j=0}^{n} \int_0^{\infty} \frac{\lambda^{k+j} y_1^{k-1} (2y_1)^j}{(k-1)! j!} e^{-2\lambda y_1} dy_1 \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{1}{2^{n+2}} \sum_{j=0}^{n} \frac{\binom{k+j-1}{k-1}}{2^{k-1}} \int_0^{\infty} \frac{\lambda^{k+j} (2y_1)^{k+j-1}}{(k+j-1)!} e^{-2\lambda y_1} dy_1 \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{1}{2^{n+2}} \sum_{j=0}^{n} \frac{\binom{k+j-1}{k-1}}{2^{k}} \int_0^{\infty} \frac{\lambda^{k+j} u^{k+j-1}}{(k+j-1)!} e^{-\lambda u} du \\
&= \frac{1}{2} - \sum_{n=0}^{k-1} \frac{1}{2^{k+n+2}} \sum_{j=0}^{n} \binom{k+j-1}{k-1} = \frac{1}{2} - \sum_{n=0}^{k-1} \frac{\binom{n+k}{n}}{2^{k+n+2}} \\
&= \frac{1}{2} - \frac{1}{4} \cdot \sum_{n=0}^{k} \frac{\binom{n+k}{n}}{2^{k+n}} + \frac{\binom{2k}{k}}{2^{2k+2}} = \frac{1}{4} + \frac{\binom{2k}{k}}{2^{2k+2}} \ .
\end{aligned} \tag{20}$$

By Eq. (11), (20) and the fact that $\Pr[E_{0<1}] = \Pr[E_{0\geq 1}]$, we conclude that

$$\phi_{\lambda,\lambda_u}(k) = 2 \cdot \Pr[E_{0<1}] = \tfrac{1}{2} + \frac{\binom{2k}{k}}{2^{2k+1}}. \qquad \square$$

A.4.1 The case $\lambda_u < \lambda$. Note that, when $\lambda_u < \lambda$, the quantity A_2 is strictly less than the r.h.s. of Eq. 18 (we can say that based on the properties of the CDF of exponential distribution). Similarly and consequently, A_3 is also strictly less than the r.h.s. of Eq. 19. From there we can deduce that $\phi_{\lambda_u,\lambda}(k) < \phi_{\lambda,\lambda}(k)$ when $\lambda_u < \lambda$.

A.5 Proof of Lemma 5

Proof. Let M_j, $j = 1, \ldots, k$ denote the event that m_x and m_y meet at the j-th hop. Further, let $Y_n = \sum_{i=0}^{n} y_i$ and $X_n = \sum_{i=0}^{n} x_i$ for $n \leq k$. We want to prove that $\Pr[\neg M] + \frac{1}{2}\Pr[M] = \phi(k)$, since:

$$\begin{aligned}
&\Pr[\neg M] + \frac{1}{2}\Pr[M] \\
&= (1 - \Pr[M]) + \frac{1}{2}\Pr[M] \qquad = 1 - \frac{1}{2}\Pr[M] \\
&= \Pr[\neg M] + \frac{1}{2}(1 - \Pr[\neg M]) \qquad = \frac{1}{2}\Pr[\neg M] + \frac{1}{2}.
\end{aligned}$$

Observe that, if the two messages do not meet they cannot swap, since,

$$\begin{aligned}
\neg M &\implies \left(\bigwedge_{i=0}^{k-1} Y_i > X_{i+1}\right) \vee \left(\bigwedge_{i=0}^{k-1} X_i > Y_{i+1}\right) \\
&\implies (Y_k > X_k \wedge y_0 > x_0) \vee (X_k > Y_k \wedge x_0 > y_0).
\end{aligned} \tag{21}$$

On the other hand, if two messages meet with each other for n times, we prove by induction that they swap with probability 0.5 for every $1 \leq n \leq k$.

We can model this with coin-toss experiments with n fair trials. Let us denote with H the case that the two messages exit the node in the opposite order (swap) than they enter the node, given that they meet in that node. Similarly, Let us denote with T the case that the two messages exit the node in the same order as they enter they node, given they meet in that node. For a general n, this random experiment will generate an n-bit string X_n. If X_n has even number of H, the messages exit the mixnode in the same order as the enter. If X_n has odd number of H, they messages will be swapped. Let S_n denote the set of all possible such strings. Further, let O_n denote the set of strings in S_n with odd number of H, and E_n denote the set of strings with even number of H.

Claim 1. *For* $1 \le n \le k$, $|O_n| = |E_n|$.

Proof of Claim . For the base case of $n = 1$, this directly follows from Lemma 1, since the two messages swap with probability 0.5. We have $S(1) = \{H, T\}$.

By inductive hypothesis, after h trials we have $|O_h| = |E_h|$. For $(h+1)$-th trial, the two messages switch their order with probability 0.5 (By Lemma 1) — and corresponds to two possible outcomes H and T. Therefore O_{h+1} will contain all the strings from O_h concatenated with T at the tail, plus all the strings from E_h concatenated with H at the tail. Similarly, E_{h+1} will contain all the strings from O_h concatenated with H at the tail, plus all the strings from E_h concatenated with T at the tail. In other words,

$$O_{h+1} = \{X||T\ \forall X \in O_h\} \cup \{X||H\ \forall X \in E_h\} \tag{22}$$
$$E_{h+1} = \{X||T\ \forall X \in E_h\} \cup \{X||H\ \forall X \in O_h\} \tag{23}$$
$$|O_{h+1}| = |O_h| + |E_h| \tag{24}$$
$$= |E_{h+1}| \tag{25}$$

where || denotes concatenation operation. And that concludes our inductive proof. ⋄

Finally, $\phi(k)$ denotes the probability that the two messages are not swapped. In particular (cf. specifically, Eq. (11) in the proof of Theorem 1), we get that:

$$\begin{aligned}
&\phi(k) = \Pr\left[\left(Y_k > X_k \wedge y_0 > x_0\right) \vee \left(X_k > Y_k \wedge x_0 > y_0\right)\right] \\
&= \Pr\left[\left(Y_k > X_k \wedge y_0 > x_0\right) \vee \left(X_k > Y_k \wedge x_0 > y_0\right) \middle| M\right] \\
&\qquad \times \Pr[M] \\
&+ \Pr\left[\left(Y_k > X_k \wedge y_0 > x_0\right) \vee \left(X_k > Y_k \wedge x_0 > y_0\right) \middle| \neg M\right] \\
&\qquad \times \Pr[\neg M] \\
&= \frac{1}{2} \cdot \Pr[M] + 1 \cdot \Pr[\neg M] = \Pr[\neg M] + \frac{1}{2}\Pr[M].
\end{aligned} \tag{26}$$

And that completes the proof of our lemma. □

A.6 Proof of Theorem 5

Proof. Analogous to the proof of Theorem 4, let us define the following two quantities:

- T is a random variable that denotes the total number of times the two challenge messages have overlapping delays on a hop. In $\mathsf{CCM}^{k,\lambda,\lambda_u}$, the two messages would meet in such a condition, however, in $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ the two messages might still end up choosing different nodes for the hop and not meet each other. If $T = 0$, the two messages definitely do not meet, and the adversary definitely wins.
- $F(t)$ denotes the probability that, for t randomly chosen hops from the path of one challenge message, other challenge message does not choose the same nodes for those hops or the node is compromised whenever they choose the same node.

Since each layer is independent of other layers in the mixnet, $F(t) = F(1)^t$. If V denotes the event that the two messages choose the same node for a given hop, and W denotes the event that the chosen node is honest,

$$F(1) = 1 - \Pr[V \wedge W] = 1 - \frac{(1-c)}{\mathsf{K}}.$$

Let us denote δ^* as the error for pairwise unlinkability provided by $\mathsf{CCM}^{k,\lambda,\lambda_u}$ when the adversary does not compromise any nodes. We know from Theorem 3 that $\delta^* = \frac{1}{2} \times \Pr[\neg M]$. For our current scenario, we can say the following about the event M' that the messages 'mix' with each other :

$$\begin{aligned}
&\Pr[\neg M'] \\
&= \Pr[T=1] \cdot F(1) + \cdots + \Pr[T=k] \cdot F(k) + \Pr[T=0] \\
&\le \Pr[T=1] \cdot F(1) + \cdots + \Pr[T=k] \cdot F(1) + \Pr[\neg M] \\
&= F(1) \cdot \Pr[M] + \Pr[\neg M] \\
&= \left(1 - \frac{(1-c)}{\mathsf{K}}\right) \cdot 2\big(1 - \phi(k)\big) + 2\left(\phi(k) - \frac{1}{2}\right).
\end{aligned} \tag{27}$$

From the above equation we can say,

$$\begin{aligned}
&\Pr\left[\mathcal{G}_{\mathsf{PU}}^{\mathsf{CCM}^{k,\lambda,\lambda_u},\mathcal{A},0}(1^\eta) = 1\right] = \frac{1}{2} + \frac{1}{2} \cdot \Pr[\neg M'] \\
&\le \frac{1}{2} + \left(1 - \frac{(1-c)}{\mathsf{K}}\right)\big(1 - \phi(k)\big) + \left(\phi(k) - \frac{1}{2}\right).
\end{aligned}$$

Therefore, the protocol $\mathsf{MCM}^{\mathsf{K},k,\lambda,\lambda_u}$ with at most c fraction of compromised nodes provides pairwise unlinkability with an error bounded by $\delta \le \left(1 - \frac{(1-c)}{\mathsf{K}}\right)\big(1 - \phi(k)\big) + \left(\phi(k) - \frac{1}{2}\right)$.

From the other direction, since we know that $F(t) = F(1)^t$, we can say

$$\begin{aligned}
&\Pr[\neg M'] \\
&= \Pr[T=1] \cdot F(1) + \cdots + \Pr[T=k] \cdot F(k) + \Pr[T=0] \\
&\ge \Pr[T=1] \cdot F(k) + \cdots + \Pr[T=k] \cdot F(k) + \Pr[\neg M] \\
&= F(k) \cdot \Pr[M] + \Pr[\neg M] \\
&= \left(1 - \frac{(1-c)}{\mathsf{K}}\right)^k \cdot 2\big(1 - \phi(k)\big) + 2\left(\phi(k) - \frac{1}{2}\right).
\end{aligned} \tag{28}$$

Therefore, error δ is lower bounded by,

$$\delta \ge \left(1 - \frac{(1-c)}{\mathsf{K}}\right)^k \big(1 - \phi(k)\big) + \left(\phi(k) - \frac{1}{2}\right)$$

□

B PRELIMINARIES ABOUT QUEUEING THEORY

Below we provide a brief summary of the relevant terms of notations related to queueing theory borrowed from [2].

Queue Networks. Queue networks are systems in which single queues are connected by a routing network. A *queue* or *queueing node* receives jobs (also called requests) that arrive to the queue, possibly wait for some time, take some time for processing, and then depart from the queue. A queue can have one or more *servers* that process the arriving jobs. When the job is completed and departs, that server will again be free to be paired with another arriving job. The number of servers in a queue represent the number of concurrent jobs the queue can process.

Queueing Node: Kendall's notation. A queueing node can be described using Kendall's notation: it uses three factors A/S/C to describe queueing models, where A denotes the time between arrivals to the queue, S the service time distribution and C the number of service channels (servers) open at the queueing node. Sometimes the notation is also extended to A/S/C/K/N/D where K is the capacity of the queue, N is the size of the population of jobs to be served, and D is the queueing discipline.

If the final three parameters are not specified (e.g. M/M/C queue), it is assumed $K = \infty$, $N = \infty$ and D = FCFS (first come first served), which is the case for us. A queue is an M/M/C queue implies that the time between arrivals follows a Markovian (M) process, i.e. the inter-arrival times follow an exponential distribution. The second M means that the service time is also Markovian: it also follows an exponential distribution. The last parameter C is the number of service channels available at the queue.

In the Context of Continuous Mixnets. We consider $K = \infty$, assuming there is no additional buffering of messages on the mixnodes, and they are processed immediately when they arrive. We can consider $N = \infty$ for a long-running system where the clients can continue to generate messages without interruptions. FCFS is the natural choice for queueing discipline considering that the delay for a packet on a mixnode is counted immediately from the time it arrives. Additionally, for our proofs, we assume that $C = \infty$: that is assuming that the system/memory limits of a mixnode is high enough to handle all the packets it receives.

Steady State of Network. A system or a network is in a steady state if the state variables that define the behaviors of the system do not change with time. For many systems, a steady state is achieved after an initial start-up or warm up period: this initial warm-up period is called a transient state.