The Last Hop Attack: Why Loop Cover Traffic over Fixed Cascades Threatens Anonymity

Maximilian Weisenseel
TU Dresden
Dresden, Germany
maximilian.weisenseel@tu-dresden.de

Christoph Döpmann
TU Berlin
Berlin, Germany
christoph.doepmann@tu-berlin.de

Florian Tschorsch
TU Dresden
Dresden, Germany
florian.tschorsch@tu-dresden.de

Abstract

Advanced mix net designs use a combination of loop cover traffic and fixed cascades to detect when active adversaries delay or drop messages. In this paper, we propose the Last Hop Attack, a new attack algorithm that takes advantage of the fact that users send loop cover, i.e., messages sent to themselves over the same mix nodes that they also use to communicate with others. We use established privacy definitions based on indistinguishability games and prove that our algorithm can break strong anonymity notions. Our research shows that the Last Hop Attack breaks Sender Receiver Pair Unlinkability for any Anonymous Communication Network that utilizes loop cover traffic, fixed cascades, and no additional cover traffic. We furthermore conclude that the notions of Sender Message Unlinkability, Receiver Message Unlinkability (and Unobservability), and Both Side Unlinkability (and Unobservability) are unachievable in this setting. To the best of our knowledge, this impossibility result is the first to show that loop cover traffic can threaten anonymity. It allows us to conclude that mix nets that utilize loop cover traffic and fixed cascades must deploy additional cover traffic to achieve strong anonymity.

Keywords

Anonymous Communication, Loop Cover Traffic

1 Introduction

In an increasingly digitized world, the need for anonymous communication has become essential, offering individuals the freedom to express themselves without fear of repercussion or surveillance. Researchers and developers have spent decades creating and analyzing Anonymous Communication Networks (ACNs) to meet these anonymity requirements. The first advances [8, 10, 30] of these ACNs offered strong privacy protection. Still, they were designed to exchange single messages and did not offer enough interactivity once the internet was used for browsing. This led to low-latency ACNs and their most prominent representative Tor [17]. Low-latency ACNs offer the performance required for interactive internet usage but cannot sufficiently protect against strong adversaries. This problem is tackled by the latest generation of ACNs, including Loopix [32] and Nym [15]. They use loop cover traffic [11] and continuous time mixing [9, 25] to achieve strong privacy notions and low latency, even against powerful adversaries. In an effort to even protect against active adversaries, Miranda [28] and SMRT [35] combine loop cover traffic with fixed cascades. The advantage of using loop messages is that the user knows exactly when each of these messages should return, so missing or delayed loop cover messages indicate an attack. Sending these loop cover messages over the same cascade as the real messages ensures that whenever an adversary delays or drops a message, they potentially tamper with a loop cover message, which will be noticed.

In this paper, we show that the combination of loop-cover traffic and fixed cascades introduces an attack vector. The main idea of this attack vector stems from a fundamental observation: Fixed cascades ensure that all messages from a given user are routed through the same mixes. Consequently, all messages originating from user A leave the network at the last mix in this cascade, i.e., their Last Hop. This includes any loop messages sent by A, which also leave the network at their Last Hop. Even if no other user communicates with A, they still send messages to themselves, which they receive back from their Last Hop. An adversary can deduce that if user A receives messages exclusively from a single mix, that mix must be their Last Hop. Furthermore, since all messages traverse the same cascade and therefore the same Last Hop, the adversary can also infer the Receiver Anonymity Set of A, which comprises all users who receive messages from this Last Hop.

We will extend this idea and prove that strong privacy notions cannot be achieved with ACNs that utilize loop cover traffic, fixed cascades, and no additional cover traffic. In the remainder of the paper, we refer to ACNs that have these properties as fixed-loop ACNs. We use the framework of Kuhn et al. [26] to prove this result. In their framework, privacy notions are defined based on indistinguishability games. These games are played within the setting of a specific ACN design. In order to break a privacy notion, the adversary has to distinguish between two scenarios. If they are able to win this indistinguishability game with a high probability, the privacy notion is not achievable.

We analyze the privacy notion of Sender Receiver Pair Unlinkability [26] (SR)$\overline{\text{L}}$. The notion describes whether an adversary can learn which sender communicates with which receiver. Therefore, we create scenarios where the senders and receivers of messages differ, but everything else is identical. Additionally, we construct a deliberately simple ACN model, the fixed-loop ACN, which complies with our three main assumptions (i. e., loop cover traffic, fixed cascades, and no additional cover traffic). We then present an attack algorithm against fixed-loop ACNs that is able to differentiate the

Proceedings on Privacy Enhancing Technologies 2025(2), 382–397

https://doi.org/10.56553/popets-2025-0067

two scenarios with high probability. We calculate this probability by identifying events that enable the adversary to determine a scenario with certainty. We then show that the adversary is able to observe these distinguishing events significantly often, which allows us to conclude that an attack algorithm exists that is able to win the (SR)$\overline{\text{L}}$ indistinguishability game on fixed-loop ACNs with a high probability. Hence, the notion of Sender Receiver Pair Unlinkability is unachievable for any fixed-loop ACN.

We prove that a global passive adversary is able to break Sender Receiver Pair Unlinkability for any fixed-loop ACN. We argue that even a partially global adversary, who can observe a subset of the mixes, is able to break this notion. Furthermore, we conclude that the notions of Sender Message Unlinkability, Receiver Message Unlinkability (and Unobservability), and Both Side Unlinkability (and Unobservability) are not achievable either.

Lastly, we show that our result is not only relevant for theoretical considerations, but also affects real mix net designs, especially SMRT and Miranda, which build on loop cover traffic and fixed cascades to detect active adversaries. This allows us to apply our impossibility result and conclude that those can only provide Sender Receiver Pair Unlinkability when utilizing cover traffic.

Our main contributions can be summarized as follows:

- We introduce the Last Hop Attack and show that it significantly diminishes the Receiver Anonymity Set of a user in the context of Miranda. This effect is amplified when the adversary is able to monitor the user for an extended period, potentially reaching a critical level (Section 2)
- We proceed by generalizing and formalizing our assumptions within an ACN model (Section 3) and outlining our attack strategy (Section 4)
- We demonstrate that our attack algorithm achieves a significant advantage against this ACN model and thereby possesses a valid attack strategy (Section 5)
- We present a novel impossibility result by proving that there cannot be an ACN that provides Sender Receiver-Pair Unlinkability and utilizes fixed cascades, loop cover traffic, and no additional cover traffic (Section 6)
- We deduce that the impossibility result extents to the notions of Sender Message Unlinkability, Receiver Message Unlinkability (and Unobservability), and Both Side Unlinkability (and Unobservability) are not achievable (Section 6)
- We analyze the real-world impact of the Last Hop Attack by discussing assumptions and consequences of our impossibility result (Section 7)

We also provide a brief discussion of related work in Section 8 before concluding our paper in Section 9.

2 The Last Hop Attack

In this section, we introduce the Last Hop Attack. We analyze the mix net Miranda from the perspective of an adversary. We describe Miranda and its threat model. Furthermore, we sketch the attack idea with an example. Afterward, we calculate the expected anonymity set size and, finally, estimate how the anonymity set shrinks over time.

2.1 Miranda

Miranda's [28] main goal is to defend against active adversaries. They assume an adversary that can observe the whole network (global passive adversary) and, at the same time, delay or drop messages at a subset of corrupted mixes (partially active adversary). We show that exactly the mechanism that protects against active attacks enables passive attacks.

Miranda uses the concept of mix nodes. Mix nodes collect multiple messages and shuffle them in order to make it more difficult to link incoming and outgoing messages. Since a single mix, of course, knows the relation between incoming and outgoing messages, multiple mix nodes are chained to form a so-called *cascade*. Each message now passes each mix node of the cascade. At fixed intervals, so-called *epochs*, the directory authorities publish a set of all currently available cascades. The users pick a cascade from this list at random and use it for the whole epoch.

Layered encryption is used to prevent an obvious correlation between incoming and outgoing packets. The sender encrypts the message, and each mix removes one layer of encryption, thereby changing the packet's binary pattern. The last layer of encryption is removed by the recipient, which is then able to read the cleartext. To this end, Miranda uses the Sphinx [3] packet format and ensures that messages are of constant length. This ensures that the messages are indistinguishable from each other at any stage in the network.

These techniques are commonly used and aim to protect against a (global) passive adversary. In order to detect active attacks, Miranda applies loop messages. As the name indicates, these messages are sent by users through the mix net back to themselves. Hence, the user knows exactly when each of these messages should return. If one loop cover message is missing or delayed, this indicates an active attack. Due to the fixed cascade, loop cover messages and real messages are sent over the same cascade. The packet format ensures that loop cover and real messages are indistinguishable. Thereby, whenever the adversary delays or drops a message, there is a chance they hit a loop cover message, for which the users can detect any delay or absence.

2.2 Threat Model

Miranda aims to achieve strong anonymity against a powerful adversary. The authors assume a global observer who is able to eavesdrop on all traffic that is exchanged in the network as well as the sending rate of users. Additionally, the adversary is able to corrupt mixes as long as the majority of mixes are honest. The adversary is able to observe the internal states and keys of all corrupted nodes. Furthermore, Miranda allows an arbitrary number of users to be malicious, as long as there exist 2ω honest users, where ω is enough to ensure that any first-mix in a cascade receives a "sufficient" number of messages to ensure reasonable anonymity. While the adversary is able to drop and delay packets on corrupted mixes, they are not able to drop packets between honest parties and can delay them only for a limited period. The goal is to hide the correspondence between senders and receivers of messages in the network. They aim to provide the same protection as an "ideal mix," i.e., a single mix node, which is known to be honest.

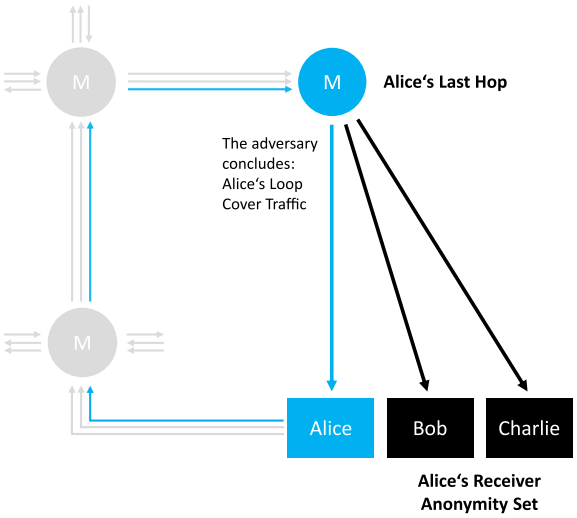

Figure 1: The Last Hop Attack.

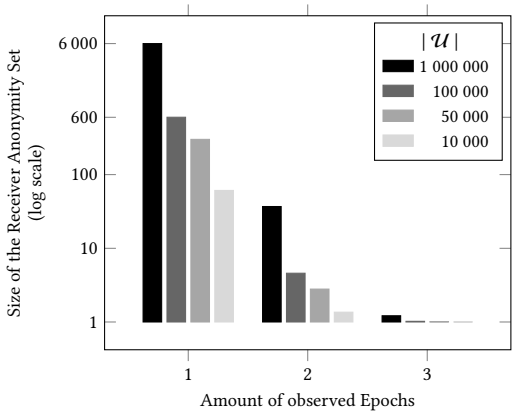

Figure 2: Size of the Receiver Anonymity Set, depending on the number of observed epochs and the number of users with $|\mathcal{M}| = 1,000$, $x = 5$ and $|A| = 1$.

2.3 The Attack Idea

Assume there is an activist leader, Alice, living in a totalitarian regime. She has a daily newsletter, which she sends to other activists: Bob and Charlie. They use Miranda to protect themselves. To protect herself, Alice keeps her identity secret so nobody can send messages to her. Nevertheless, the regime already identified her but is much more interested in the people she communicates with. They want to find the other activists she is sending messages to; in anonymity terms, her Receiver Anonymity Set.

We take the position of the adversary with the capabilities defined in Miranda's threat model: a global passive adversary. Figure 1 illustrates the network from the perspective of the adversary and highlights the Last Hop Attack. They observe the communication between the mixes and users but not the content of the messages. From this perspective, the adversary can identify the Last Hop of Alice and, thereby, her Receiver Anonymity Set. They can do this with the help of three observations. First, Alice sends all of her messages through her chosen cascade. Thus, all her messages pass through the same mixes and leave at the same Last Hop. Second, if all her messages leave at the same Last Hop, this Last Hop also sends her loop cover messages back to her. The adversary does not know which cascade Alice has chosen, but they can observe that exactly one mix is sending messages to Alice (blue arrow), which then has to be her Last Hop. Third, since it is her Last Hop, the adversary can conclude that all other messages Alice sends are routed through this Last Hop, and thereby, her Receiver Anonymity Set consists of all users who received messages from this mix.

2.4 Expected Anonymity Set Size

We aim to estimate Alice's Receiver Anonymity Set by determining the number of users who receive messages during a single epoch from a specific Last Hop. First, we calculate the number of users who select the same Last Hop as Alice. For this initial estimation, we assume that users choose their Last Hops uniformly from the set of all mixes. Consequently, the expected number of users selecting a particular Last Hop depends solely on the total number of active users in the network, denoted as $|\mathcal{U}|$, and the number of mixes, denoted as $|\mathcal{M}|$. Thus, the expected number of users per Last Hop is $\frac{|\mathcal{U}|}{|\mathcal{M}|}$. Next, we need to consider the number of communication partners. This factor significantly influences the anonymity set. If users do not actively send messages, only their cover traffic contributes to the anonymity set. If they send in addition to their cover traffic to x other users, these are then also included in the anonymity set. Note that this approach might overestimate the anonymity set. If user X and user Y send messages to the same user Z, Z will be counted twice. Therefore, this calculation should be considered as an upper bound for the Receiver Anonymity Set. We refer to the number of users Alice actually communicates with as $|A|$ and compute Alice's expected Receiver Anonymity Set.

$$R_{AS} = \frac{|\mathcal{U}|}{|\mathcal{M}|} \cdot (x+1) + |A|$$

First, it is important to note that observing the Last Hop is sufficient to determine the Receiver Anonymity Set. This is not the case for other ACNs. For instance, determining the Receiver Anonymity Set in the Tor network requires tracing the cascade and considering all potential paths the message may have taken. Additionally, since monitoring the Last Hop is enough to identify the Receiver Anonymity Set, other network parameters, such as path length, do not affect the attack.

When considering the parameters of the formula, the Receiver Anonymity Set of a user depends on the number of other active users, the number of mixes in the network, and the number of communication partners. We can observe that increasing numbers of users and communication partners increases the anonymity set, while more mixes and, thereby, more potential Last Hops decrease the anonymity set.

2.5 Multiple Epochs

In certain situations, the adversary may be able to correlate their observations across multiple epochs. This is feasible because, within each time period, the adversary can ascertain Alice's Last Hop and,

consequently, her Receiver Anonymity Set. We consider our previous example to illustrate this. Here Alice sent a daily newsletter. If this is known by the adversary, they are able to intersect the anonymity sets of multiple epochs and narrow down the potential receivers. Given that her subscribers receive a message from her every day, other users can only contribute to the anonymity set if they also receive messages from the respective Last Hop that Alice has chosen in all considered epochs.

There are three possibilities for a user of the initial anonymity set R_{AS} to remain in the anonymity set. They choose the same Last Hop as Alice, one of their communication partners chooses the same Last Hop as Alice, or they receive a message from Alice (|A|). Thereby if they do not receive a message from Alice, they or one of their communication partners need to choose the same Last Hop as Alice in every epoch in order to contribute to the anonymity set. We start with the initial set and then multiply this by the probability that these users remain in the anonymity set. This allows us to model the expected anonymity set size after $|E|$ epochs. Here x is again the number of users a user sends messages to and $|A|$ the number of users Alice sends messages to.

$$R_{AS}^{|E|} = \frac{1}{|\mathcal{M}|} \cdot |\mathcal{U}| \cdot (x+1) \cdot \left(1 - \left(1 - \frac{1}{|\mathcal{M}|}\right)^{(x+1)}\right)^{|E|} + |A|$$

Figure 2 illustrates the expected anonymity set size on the y-axis (in log scale) depending on the number of observed epochs (x-axis) for a different number of users $|\mathcal{U}|$ (bars). We assume a network with 1,000 mixes. Additionally, we need to make assumptions about the traffic of other users. We assume that each user sends messages to five other users ($x = 5$), which we presume is realistic. However, as long as x remains small ($x < 100$), we can observe similar effects since then the anonymity sets scales linear with the size of x. Finally, we assume that Alice communicates with one other user. If the adversary observes only one epoch, an increased number of users has a significant effect on the anonymity set. However, this advantage diminishes rapidly if the adversary can correlate multiple epochs. While some of the anonymity set sizes after two epochs may be considered large enough. The expected anonymity set after three epochs is in all cases, close to 1; the number of users to whom Alice actually sends messages. Even with 1 million concurrent users, the expected anonymity set after three epochs is 1.2.

2.6 Multiple Potential Last Hops

In this example, we assumed Alice does not receive messages from other users. Therefore, we want to briefly cover the adversary's strategy if Alice receives messages from multiple mixes. In this case, it is not clear which of the mixes sent to Alice is her Last Hop. Therefore, the anonymity set encompasses all users who receive messages from the same mixes as Alice. So if Alice receives messages from mix M_1 and M_3, all users who receive messages from M_1 or M_3 might have received a message from Alice.

3 Formalizing the Attack

After outlining the potential of the attack, we proceed with an in-depth analysis using a formal framework [26]. This allows us to clearly define our assumptions and to prove our results for an abstract model that can be applied to different ACNs. Additionally, the framework enables us to precisely measure the strength of the adversary and derive clear boundaries; even in the style of differential privacy [19] (see Appendix B). It also provides hierarchical relations between the privacy notions, allowing us to break one of the weakest notions and thereby demonstrate that most of the other notions are also not attainable. As the framework assumes a very strong adversary model, we limit the capabilities of this adversary to ensure that our results are applicable to real-world scenarios.

3.1 The Fixed-Loop ACN Model

We consider a general ACN with three defining properties: fixed cascades, loop cover traffic, and no additional cover traffic.

We say a *cascade* is a sequence of mixes a user uses to forward their message through the network. We use the term *fixed cascade* to define that a client uses the same sequence of mix nodes for all of their messages. They might change their selection in a regular interval, which we refer to as *epoch* [2, 15].

We consider all messages as *cover traffic* that are sent with the goal of obfuscation and do not contain any information users want to exchange with each other. In particular, this can be messages that are sent to randomly selected other users or mix nodes and then dropped at reception. This includes the mechanism that Loopix [32] uses between providers and users and any other messages that are used for obfuscation without delivering information. However, communications from other clients that might naturally populate the network are not considered cover traffic.

We say *loop cover traffic* [11, 32] for all messages a user sends over the chosen cascade to themselves to detect active adversaries.

This model allows us to define our three main assumptions:

A_1 : Fixed cascades for the duration of an epoch
A_2 : Every user sends at least one loop message per epoch
A_3 : No cover traffic, except loop cover traffic

In the following, when referring to this setting, we denote it as *fixed-loop ACN* and accordingly, imply the respective set of assumptions.

Since we assume fixed cascades, we also need to consider the cascade selection, i. e., which nodes are included in a specific cascade. For our calculations, we assume a uniform cascade selection, where the probability of being selected in a cascade and their position are equal for all nodes. We discuss and relax this assumption in Subsection 7.1. Additionally, we demonstrate in Appendix C that our attack is also feasible with a bandwidth-based cascade selection, where nodes are selected based on the bandwidth they provide. For now, we make the following additional assumption:

A_4 : Nodes and their position in the cascade are drawn uniformly at random from all nodes, and users select their cascade uniformly at random

3.2 The Game

The framework of Kuhn et al. [26] defines indistinguishability games to prove whether a given privacy notion is achievable for a given ACN. Instead of measuring anonymity sets, the anonymity provided by an ACN is measured in how difficult it is for the adversary to differentiate two almost identical scenarios. The probability of distinguishing the scenarios is measured as *advantage*. If the adversary is able to achieve a non-negligible advantage in the game of a given notion, the adversary must have a valid strategy, and thereby, the privacy notion is not achievable.

Table 1: Communication batches for Mix Sender Receiver along the lines of Kuhn et al. [26]

Instance	Scenario	
	$b = 0$	$b = 1$
$a = 0$	$u_0 \longrightarrow u_A$ $u_1 \longrightarrow u_B$	$u_0 \longrightarrow u_B$ $u_1 \longrightarrow u_A$
$a = 1$	$u_1 \longrightarrow u_B$ $u_0 \longrightarrow u_A$	$u_1 \longrightarrow u_A$ $u_0 \longrightarrow u_B$

These games consist of a challenger, an ACN model, and an adversary. The adversary is the only real player. They win the game if they can differentiate two scenarios. A scenario defines the communication happening, i.e., which user sends messages to which other user. The adversary can freely choose the two scenarios. However, the chosen scenarios have to comply with certain rules. For example, sending a message in both scenarios, as otherwise, it might be possible to win the game without any actual capabilities. In this paper, we focus on the notion of Sender Receiver Unlinkability. It describes if an adversary is able to identify which sender sent a message to which receiver.

We chose this notion for two reasons. First, it is the notion Miranda identifies as their security goal - they want to hide the correspondence between senders and recipients of the messages. Second, it is one of the weakest notions defined in the framework, and therefore, proving that it is not achievable demonstrates that most other notions are also not achievable. The framework of Kuhn et al. [26] already analyses which properties the scenarios have to provide to achieve this notion. Therefore, we refer the interested reader to their definition and continue with the construction.

The adversary selects four users u_0, u_1, u_A, and u_B at random. They send two identical messages in each scenario. In scenario $b = 0$, they are sent from u_0 to u_A and from u_1 to u_B. In scenario $b = 1$, from u_0 to u_B and from u_1 to u_A. To comply with the rules, the adversary has to provide two instances for each scenario. In the first scenario, u_0 sends first; in the second scenario, u_1 sends first.[1] See Table 1 for a summary of the scenario-instance combinations. This choice of scenarios and instances complies with the definition of *Mix Sender Receiver* M_{SR}, E_{SR}, and $\not\Diamond$ from [26] and thereby fulfills the requirements that allow us to analyze Sender Receiver Pair Unlinkability (SR)$\overline{\text{L}}$. We can now use the constructed scenarios to play the indistinguishability game. First, the challenger randomly selects the instance bit a and the challenge bit b. Then, the adversary sends a query to the challenger. This query contains the constructed two scenarios with two instances each. The challenger checks if the query conforms with the chosen notion. If they comply, the challenger selects scenario b and instance a and simulates it on the given ACN. The output of this simulation is then returned to the adversary. Finally, the adversary has to submit a guess g for the scenario bit b. The adversary wins if $g = b$.

[1] See Section 4 of Kuhn et al. for details on why this is necessary.

3.3 The Advantage

We can measure the strength of the adversary by calculating the probability of $g = b$ given b with:

$$\Pr(g = b) = \frac{1}{2}\Pr(g = 1|b = 1) + \frac{1}{2}\Pr(g = 0|b = 0)$$

Note that the adversary can win, in expectation, half of the games by simply guessing. So, to measure the strength of the adversary, we want to analyze how much better they can do than simply guessing. We call this the advantage of the adversary α.

Intuitively, we can calculate the advantage by considering the adversary's success probability minus the probability of *simple guessing* $\frac{1}{2}$. This leaves us with a range of $[0; \frac{1}{2}]$, which we can normalize to the more intuitive range of $[0; 1]$ by multiplying by two. An advantage of zero represents that the best the adversary can do is guessing. An advantage of one indicates that the adversary has a strategy that enables them to win the game every time. We use the definition of Kuhn et al.:[2]

$$\alpha = |\Pr(g = 0|b = 0) - \Pr(g = 0|b = 1)|$$

Based on the advantage, we can now define whether a privacy notion is achievable. If the adversary can win the game with a probability higher than simple guessing, they must have a valid strategy, and the notion is not achievable. Note that this is identical to the adversary having a non-negligible advantage. An ACN achieves a notion if no probabilistic polynomial time algorithm (PPT) achieves a negligible advantage in the indistinguishability game.[3]

3.4 Restricting the Adversary

The chosen framework analyzes a worst-case scenario in which the adversary can control all communication in the network as long as it complies with the rules of a given notion. Note that for our chosen notion, this would mean that only u_0 and u_1 sending messages through an otherwise empty network would be a valid scenario. While this might be interesting for theoretical analysis, achieving even the weakest notion, (SR)$\overline{\text{L}}$, would be challenging for any practical ACN.

In order to analyze a more realistic setting, we restrict the capabilities of the adversary and allow all users to send arbitrarily many messages to all other users. The only communication under the adversary's control are the messages sent and received by the challenge users. We require them to only send the messages defined by the notion of Sender Receiver Pair Unlinkability, as defined above, and the loop cover messages induced by the protocol. Additionally, we restrict that no other user sends messages to them.

A_5 : The only users sending to the challenge users are the challenge users themselves.

This restricts the communication of the four users targeted by the adversary. Note, if we would not exclude the challenge users, "ultimate anonymity" can be achieved with a simple protocol, where each user sends in every round one message to each other user [22].

[2] We can show that $(P(g = b) - 0.5) \cdot 2 = (\frac{1}{2} \cdot P(1|1) + \frac{1}{2} \cdot P(0|0) - 0.5) \cdot 2 = P(1|1) - P(0|0) - 1 = 1 - P(0|1) + P(0|0) - 1 = P(0|1) + P(0|0)$. Note this is not entirely identical. In Kuhn's definition, an algorithm that always guesses wrong also has an advantage of 1. This makes sense, given an algorithm that always guesses wrong, it is trivial to build an algorithm that always guesses right.

[3] See Kuhn et al. for the formal definition.

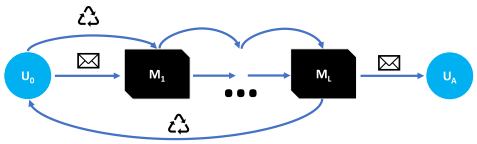

Figure 3: Fixed cascade of user u_0 (in b = 0).

Since a user's privacy should not depend on the number of exchanged messages or the number of communication partners, it is valid to assume the worst case.

4 The Attack Strategy

The adversary's strategy is based on the idea of distinguishing events. If the adversary witnesses one of these events, they are able to identify the scenario with certainty. We first give an example of such an event. Then, identify multiple other events and combine them into an attack algorithm.

4.1 Distinguishing Events

We identify distinguishing events in the setting described above. This implies that the simulated scenario-instance combination is one of the four provided by the adversary, and the ACN on which the game is played complies with our assumptions.

None of the scenarios contains a message to u_0 and we excluded cover traffic (A_3) as well as messages from other users to u_0 (A_5) in our ACN model, thereby the only way u_0 receives a message is if they sent a loop cover message. Thus, only one user sends to u_0, which is u_0 itself. Since we assumed fixed cascades, all loop cover messages of a user take the same path. Therefore, only one mix sends to u_0; the Last Hop in its cascade. See Figure 3 for an illustration. We refer to the first mix in a user's cascade as M_1 and the last mix as M_L. We know from our assumptions that u_0 sends loop cover traffic. Furthermore, we know from the provided scenarios that u_0 sends a real message to either u_A or u_B. We know that the real message and the loop cover message use the same cascade since we assumed fixed cascades in our ACN model. We have already noticed that there is only one way for u_0 to receive messages, and those are loop messages. Consequently, only one Last Hop sends to u_0 and this same Last Hop must also have sent the challenge message to either u_A or u_B.

With these observations in mind, consider a mix M with the following communication relations. M sends a message to u_0, and it sends a message to u_A. It does not send a message to u_B. The mix sent a message to u_0; therefore, we know it is the Last Hop in the cascade chosen by u_0 ($M = M_L$). Additionally, we know that it must have forwarded the challenge message. Either to u_A or u_B. We also observed that M did not send a message to u_B, so u_0 must have sent the message to u_A, which reveals to the adversary that they are in scenario $b = 0$. Note that the existence and observability of this distinguishing event depends only on the Last Hops of the challenge users. We analyze the outgoing messages to the challenge users. As long as the challenge users do not send or receive messages from other users it is irrelevant how many other users chose the same Last Hop and how many messages they might send.

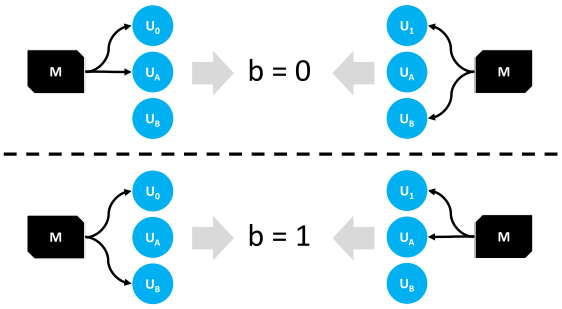

Figure 4: Visual representation of the distinguishing events.

4.2 Further Distinguishing Events

We established that the adversary can detect the scenario $b = 0$ if the Last Hop of user u_0 sends only to u_A and not to u_B. With the same idea, they can detect scenario $b = 1$ if a mix sends messages to u_0, u_B but not to u_A. Since it sends to u_0, it has to be the Last Hop in the cascade chosen by u_0, which has also forwarded the challenge message. Since the mix did not send any message to u_A, it must have sent it to u_B, which ensures the adversary that the scenario is $b = 1$. Analogously, the adversary can analyze the Last Hop of user u_1. If u_1's Last Hop sends only to u_A, they can identify scenario $b = 1$, and if it sends only to u_B, they can recognize scenario $b = 0$. See Figure 4 for a visualization of all four cases.

4.3 The Algorithm

The attack algorithm is based on the distinguishing events described above. It considers all corrupted mixes, and if a distinguishing event is observed on one of them, it can determine the scenario with certainty. If it cannot identify the scenario on any of the corrupted mixes, it will just guess the scenario.[4]

Note that in this setting, the adversary only has information about the mixes they can observe. We refer to the set of corrupted mixes as Ψ and to the observation of these mixes as O_Ψ.

5 Calculating the Advantage

In the framework of Kuhn et al., a privacy notion is not achievable when an attack algorithm can win the indistinguishability game of the chosen notion on the chosen ACN with a non-negligible advantage. We already presented two scenarios that comply with the Sender Receiver Pair Unlinkability notion and a general ACN that is only defined by our three assumptions of fixed cascades, loop cover traffic, and the fact that it sends no additional cover traffic.

In the following, we show that the adversary's advantage is non-negligible, which eventually leads to our impossibility result. We showed in the previous section that the adversary is able to determine the scenario correctly when it observes a distinguishing event. It is therefore enough to show that the probability of the adversary witnessing a distinguishing event is non-negligible.

Therefore, we divide the possible *user behavior* into five disjoint sets, the *user behaviors*. Based on these sets, we calculate the number

[4]We *could* improve this result, for example, with [23]. Based on the observed data, they guess which scenario is more likely. Although it provides better results, it makes the analysis more complex, so we do not use it at this point and merely point out that our results should be interpreted as lower bounds.

Algorithm 1 Algorithm for the Last Hop Attack

Require: O_Ψ {The output the challenger computed}
1: **for** $m \in \Psi$ **do**
2: **if** $m \rightarrow u_0 \wedge m \rightarrow u_A \wedge m \not\rightarrow u_B$ **then**
3: return 0
4: **end if**
5: **if** $m \rightarrow u_0 \wedge m \rightarrow u_B \wedge m \not\rightarrow u_A$ **then**
6: return 1
7: **end if**
8: **if** $m \rightarrow u_1 \wedge m \rightarrow u_A \wedge m \not\rightarrow u_B$ **then**
9: return 1
10: **end if**
11: **if** $m \rightarrow u_1 \wedge m \in m \rightarrow u_B \wedge m \not\rightarrow u_A$ **then**
12: return 0
13: **end if**
14: **end for**
15: return UniformlyRandom([0,1])

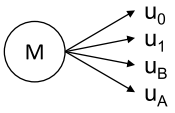

Figure 5: Representation of u_0 and u_1 choosing the same Last Hop.

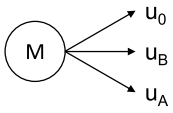

Figure 6: Representation of u_0 and u_B choosing the same Last Hop.

of distinguishing events for each of the five behaviors and the probability of each of them happening. Finally, we use these results to calculate the adversary's probability of observing a distinguishing event and show that this is equivalent to the advantage.

5.1 User Behavior

We already noticed that the only mixes where the adversary can observe our defined distinguishing events are the Last Hops of u_0 and u_1. Note that even if the adversary can control the scenarios, they cannot control the users' behavior. Therefore, they do not know which user selects which cascade. However, cascade selection affects the existence of distinguishing events. Consider Figure 5. If user u_0 and u_1 select the same Last Hop M, this Last Hop sends loop cover messages to u_0 and u_1, as well as real messages to u_A and u_B. This observation is identical for both scenarios $b = 0$ and $b = 1$. Therefore, the adversary cannot distinguish the scenarios.

But even if u_0 and u_1 choose different Last Hops, it is not guaranteed that a distinguishing event is observable on their Last Hops. Consider Figure 6. Assume scenario $b = 0$, u_0 sends a real message to u_A. If u_B selects a cascade that uses the same Last Hop as u_0, this Last Hop sends messages to u_0, u_A, and u_B. Therefore, the adversary is not able to observe a distinguishing event there. We can make an analogous argument for u_1.

After introducing these three cases, we divide all possible cascade selections that a user could make into five categories. We call them *user behaviors*, which we illustrate in Figure 7. The upper row refers to the Last Hop chosen by u_0 and the lower row refers to the Last Hop chosen by u_1. The columns describe the different user behaviors that we want to consider. We indicate the absence of a distinguishing event by a black circle and use a blue circle when a distinguishing event is observable.

Table 2: Notation Used to Describe User Behavior

Notation	Description
$\mathcal{L}(x, y)$	User x and User y chose the same Last Hop
$\overline{\mathcal{L}}(x, y)$	User x and User y chose different Last Hops
S_i	A predicate indicating that Situation i has occurred
D_{S_*}	Number of Last Hops on which the adversary could witness a distinguishable event given situation S_i has occurred

In the first user behavior (B_1), u_0 and u_1 chose different Last Hops. Additionally, u_A chose a Last Hop different from u_1 and u_B chose a Last Hop different from u_0. Consequently, two distinguishing events are observable, one on the Last Hop of u_0 and one on the Last Hop of u_1. In user behaviors B_2 and B_3, both users u_0 and u_1 choose different Last Hops. However, in B_2, user u_A chooses the same Last Hop as user u_1 and in B_3, user u_B chooses the same Last Hop as u_0. This implies that only one distinguishing event can be observed in both of these cases. In B_4, users u_0 and u_1 choose different Last Hops, but u_B chooses the same Last Hop as u_0 and u_A chooses the same Last Hop as u_1. As a result, no distinguishing event can be observed in this case. Similarly, in B_5, where u_0 and u_1 chose the same Last Hop, no distinguishing event can be observed.

In scenario $b = 1$, we can make similar observations for user behavior B_1 to B_5. In this scenario, u_0 sends a challenge message to u_B, and u_1 sends its challenge message to u_A. Therefore, we can observe a distinguishing event on the Last Hop of u_0 if u_0 selects a Last Hop that differs from both u_1 and u_A. Similarly, we can observe a distinguishing event on the Last Hop of u_1 if it chooses a Last Hop that differs from both u_0 and u_B.

Before formalizing these user behaviors, we introduce additional notation to maintain readability during this section. We use $\mathcal{L}(x, y)$ to indicate that x and y chose the same Last Hop and $\overline{\mathcal{L}}(x, y)$ to indicate that x and y chose different Last Hops. Additionally, we use B_i to refer to a specific user behavior and D_{B_i} to refer to the number of distinguishable events the user behavior B_i enables. With this notation, we can now define the situations:

$$B_1 = \overline{\mathcal{L}}(0,1) \wedge \overline{\mathcal{L}}(b,B) \wedge \overline{\mathcal{L}}(1-b,A) \qquad D_{B_1} = 2$$
$$B_2 = \overline{\mathcal{L}}(0,1) \wedge \overline{\mathcal{L}}(b,B) \wedge \mathcal{L}(1-b,A) \qquad D_{B_2} = 1$$
$$B_3 = \overline{\mathcal{L}}(0,1) \wedge \mathcal{L}(b,B) \wedge \overline{\mathcal{L}}(1-b,A) \qquad D_{B_3} = 1$$
$$B_4 = \overline{\mathcal{L}}(0,1) \wedge \mathcal{L}(b,B) \wedge \mathcal{L}(1-b,A) \qquad D_{B_4} = 0$$
$$B_5 = \mathcal{L}(0,1) \qquad D_{B_5} = 0$$

We defined the user behaviors and determined the number of distinguishing events that occur in each of them. To calculate the advantage, we first determine the probability of an adversary witnessing a distinguishing event in each user behavior and then the probability of each of these user behaviors happening. We can calculate the total probability of the adversary witnessing a distinguishing event with these probabilities.

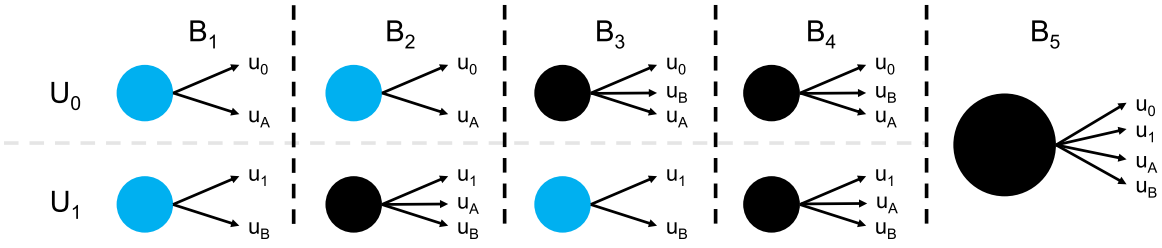

Figure 7: User behavior for $b = 0$.

5.2 Conditional Probability of Observing Distinguishing Events

Since the adversary in our model observes only a subset of the mixes, it is not guaranteed that they witness every distinguishing event that occurs. We, therefore, are interested in the probability of an adversary witnessing a distinguishing event based on the number of corrupted mixes they can observe.

The probability of the adversary witnessing a distinguishing event depends on the number of mixes the adversary has corrupted $|\Psi|$ and the total number of mixes $|\mathcal{M}|$. We model this as drawing $|\Psi|$ of the $|\mathcal{M}|$ mixes without replacement. Depending on the user behavior, there are $D_{B_i} : i \in [1, 5]$ *successful* draws, in which the adversary draws a distinguishing mix. We can use the hypergeometric distribution [34] to calculate the probability of drawing a distinguishing mix based on user behavior.

$$\Pr(x = k|B_i) = \frac{\binom{D_{B_i}}{k}\binom{|\mathcal{M}| - D_{B_i}}{|\Psi| - k}}{\binom{|\mathcal{M}|}{|\Psi|}}$$

We can calculate the probability of drawing no distinguishing mix in behavior i with $k = 0$ Consequently, the probability of drawing at least one distinguishing mix is:

$$\Pr(x \geq 1|B_i) = 1 - \frac{\binom{|\mathcal{M}| - D_{B_i}}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}} \tag{1}$$

5.3 Probability of the User Behavior

We calculate the probability of the user behaviors with a probability tree, which is depicted in Figure 8. Firstly, this enables us to directly deduce the probabilities of the different sets $\Pr(B_i)$. Secondly, it shows that our defined user behaviors are disjoint, and that their union is complete. In order to calculate the probabilities of the branches, we assume that users select their cascades uniformly at random. Note that this assumption is not strictly necessary[5]. Consequently, a user's probability of selecting a specific Last Hop is $\frac{1}{|\mathcal{M}|}$. With this assumption, we can determine the probability of the defined user behaviors. Thereby, the probability of a user selecting a specific mix as their Last Hop is $\frac{1}{|\mathcal{M}|}$ and the probability of two users selecting the same Last Hop is $\frac{1}{|\mathcal{M}|}$. We can read off the probabilities from the probability tree:

$$\Pr(B_1) = \Pr(\overline{\mathcal{L}}(0,1) \wedge \overline{\mathcal{L}}(b,B) \wedge \overline{\mathcal{L}}(1-b,A)) = \left(1 - \frac{1}{|\mathcal{M}|}\right)^3$$

$$\Pr(B_2) = \Pr(\overline{\mathcal{L}}(0,1) \wedge \overline{\mathcal{L}}(b,B) \wedge \mathcal{L}(1-b,A)) = \left(1 - \frac{1}{|\mathcal{M}|}\right)^2 \cdot \frac{1}{|\mathcal{M}|}$$

$$\Pr(B_3) = \Pr(\overline{\mathcal{L}}(0,1) \wedge \mathcal{L}(b,B) \wedge \overline{\mathcal{L}}(1-b,A)) = \left(1 - \frac{1}{|\mathcal{M}|}\right)^2 \cdot \frac{1}{|\mathcal{M}|}$$

$$\Pr(B_4) = \Pr(\overline{\mathcal{L}}(0,1) \wedge \mathcal{L}(b,B) \wedge \mathcal{L}(1-b,A)) = \left(1 - \frac{1}{|\mathcal{M}|}\right) \cdot \left(\frac{1}{|\mathcal{M}|}\right)^2$$

$$\Pr(B_5) = \Pr(\mathcal{L}(0,1)) = \frac{1}{|\mathcal{M}|}$$

5.4 Total Probability of Observing Distinguishing Events

With the probability of a certain user behavior $\Pr(B_i)$ and the probability that the adversary observes a distinguishing event depending on the user behavior, we can calculate the total probability of an adversary witnessing a distinguishing event $\Pr(\mathcal{D}_p)$:

$$\Pr(\mathcal{D}_p) = \sum_{i=1}^{5} \Pr(x \geq 1|B_i) \cdot \Pr(B_i)$$

This sum can be dissolved, and we can insert the probabilities for corrupting at least one distinguishing Last Hop (Equation 1):

$$\begin{aligned}\Pr(\mathcal{D}_p) = &\left(1 - \frac{\binom{|\mathcal{M}|-2}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \Pr(B_1) + \left(1 - \frac{\binom{|\mathcal{M}|-1}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \Pr(B_2) \\ &+ \left(1 - \frac{\binom{|\mathcal{M}|-1}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \Pr(B_3) + \left(1 - \frac{\binom{|\mathcal{M}|-0}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \Pr(B_4) \\ &+ \left(1 - \frac{\binom{|\mathcal{M}|-0}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \Pr(B_5)\end{aligned}$$

[5] See Section 7 for a discussion of the assumptions and Appendix C, where we calculate the probabilities for a bandwidth-weighted cascade selection.

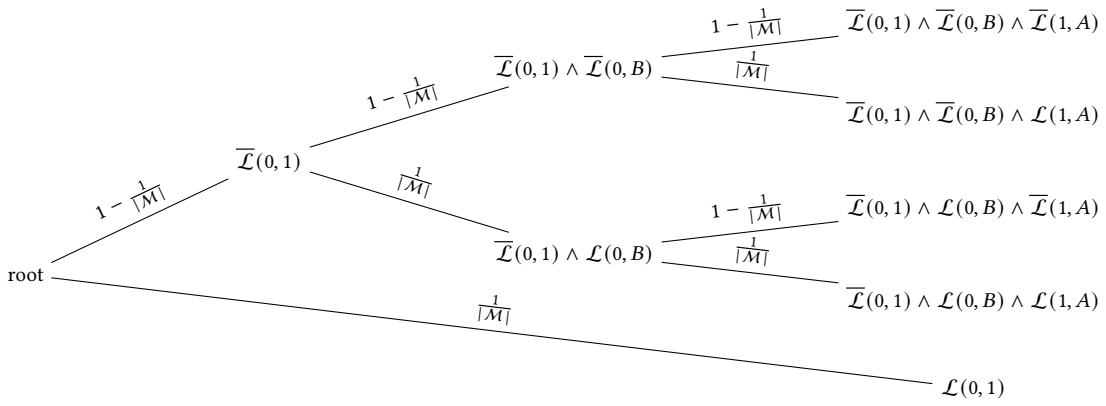

Figure 8: Probability tree for B_1 to B_5.

Since B_4 and B_5 have no and B_2 and B_3 have the same amount of distinguishing events, the equation can be simplified to:

$$\Pr(\mathcal{D}_p) = \left(1 - \frac{\binom{|\mathcal{M}|-2}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \Pr(B_1) + \left(1 - \frac{\binom{|\mathcal{M}|-1}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot (\Pr(B_2) + \Pr(B_3))$$

Finally, we can insert the previously calculated probabilities.

$$\begin{aligned}\Pr(\mathcal{D}_p) = &\left(1 - \frac{\binom{|\mathcal{M}|-2}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \left(1 - \frac{1}{|\mathcal{M}|}\right)^3 \\ &+ \left(1 - \frac{\binom{|\mathcal{M}|-1}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot 2 \cdot \left(1 - \frac{1}{|\mathcal{M}|}\right)^2 \cdot \frac{1}{|\mathcal{M}|}\end{aligned} \tag{2}$$

Equation 2 allows us to calculate the probability of an adversary witnessing a distinguishing event based on the amount of corrupted mixes $|\Psi|$ and the number of total mixes $|\mathcal{M}|$.

5.5 The Advantage

The previous formula calculates the probability of the adversary witnessing a distinguishing event. We will show that this is equivalent to the adversary's advantage. The advantage is defined as:

$$\alpha = |\Pr(g = 0|b = 0) - \Pr(g = 0|b = 1)|$$

We consider $\Pr(g = 0|b = 0)$ first. We already argued that the adversary's guesses are always correct when they witness a distinguishing event. Additionally, we specified that they toss a coin if they cannot observe a distinguishing event.

$$\Pr(g = 0|b = 0) = \Pr(\mathcal{D}_p) \cdot 1 + (1 - \Pr(\mathcal{D}_p)) \cdot \frac{1}{2}$$

We can argue similarly for $\Pr(g = 0|b = 1)$. When they witness a distinguishing event, they will never guess $b = 0$; if they do not, they toss a coin.

$$\Pr(g = 0|b = 1) = \Pr(\mathcal{D}_p) \cdot 0 + (1 - \Pr(\mathcal{D}_p)) \cdot \frac{1}{2}$$

With these two probabilities, we can calculate the advantage:

$$\begin{aligned}\alpha =& |\Pr(g = 0|b = 0) - \Pr(g = 0|b = 1)| \\ =& \Pr(\mathcal{D}_p) \cdot 1 + \left(1 - \Pr(\mathcal{D}_p)\right) \cdot \frac{1}{2} \\ &- (\Pr(\mathcal{D}_p) \cdot 0 + \left(1 - \Pr(\mathcal{D}_p)\right) \cdot \frac{1}{2}\Big) \\ =& \Pr(\mathcal{D}_p) + (1 - \Pr(\mathcal{D}_p)) \cdot \frac{1}{2} - (1 - \Pr(\mathcal{D}_p)) \cdot \frac{1}{2} \\ =& \Pr(\mathcal{D}_p) \\ =& \left(1 - \frac{\binom{|\mathcal{M}|-2}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot \left((1 - \frac{1}{|\mathcal{M}|}\right)^3 \\ &+ \left(1 - \frac{\binom{|\mathcal{M}|-1}{|\Psi|}}{\binom{|\mathcal{M}|}{|\Psi|}}\right) \cdot 2 \cdot \left(1 - \frac{1}{|\mathcal{M}|}\right)^2 \cdot \frac{1}{|\mathcal{M}|}\end{aligned} \tag{3}$$

We can conclude that the adversary's advantage is $\Pr(\mathcal{D}_p)$.

6 The Impossibility Result

So far, we defined an abstract ACN that complies with our three main assumptions of loop cover traffic, fixed cascades, and no additional cover traffic as well as two scenarios that comply with the rules of the Sender Receiver Pair Unlinkability privacy notion. We presented an attack algorithm based on distinguishing events that is able to differentiate the two scenarios with a high probability.

In order to break the notion of Sender Receiver Pair Unlinkability, we need to show that the adversary's advantage is non-negligible. Breaking the Sender Receiver Pair Unlinkability leads eventually to the impossibility result.

We evaluate the adversary's advantage by analyzing Equation 2. It depends on two variables: the total number of mixes $|\mathcal{M}|$ and the number of corrupt mixes $|\Psi|$. We present results for some parameters but encourage calculating different combinations. Please feel free to experiment with the code we provide in Appendix A.

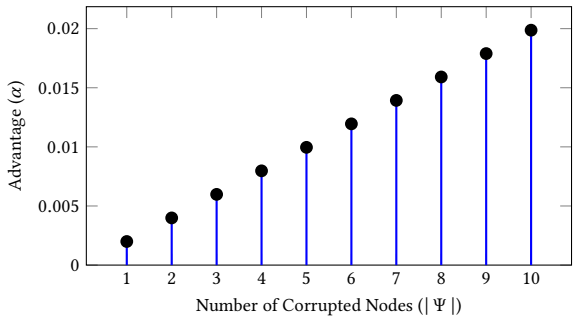

Figure 9: Advantage of the adversary depending on the number of corrupted nodes for $|\mathcal{M}| = 1,000$.

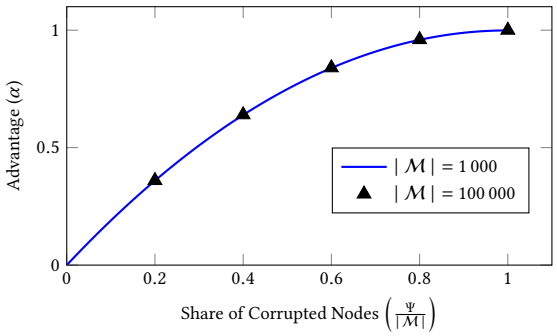

Figure 10: Advantage of the adversary depending on the proportion of the corrupted nodes.

In Figure 9, we depict the number of corrupted nodes on the x-axis and the adversary's advantage on the y-axis. The results show that even if the adversary observes only a single node of the 1,000 nodes in total, they can already gain an advantage of ≈ 0.002. For ten observed nodes, the adversary's advantage rises to ≈ 0.02.

In Figure 10, we depict the adversary's advantage (y-axis) in comparison to the relative share of corrupted nodes (x-axis). A relative share of zero corrupted nodes yields an entirely trustworthy network. Accordingly, a relative share of one yields a network where the adversary can observe every node, which corresponds to a global passive adversary. When the adversary is able to observe a tenth of the nodes, their advantage is ≈ 0.19. Considering the global passive adversary, the advantage converges to ≈ 0.999.

From our results, we can certainly conclude that the adversary's advantage is non-negligible for the global passive adversary. For weaker adversaries, we argue that the adversary's advantage is still non-negligible. We leave the decision at which exact point the adversary's advantage can be considered no longer negligible to the reader. Please note, however, that for prolonged observation periods (i. e., multiple epochs), an adversary can accumulate its advantage. The embedding of epochs in the protocol and, thereby, the natural repetition of the game is a valuable setting for the adversary. Every epoch is a new chance for the adversary to observe a distinguishing event and to deanonymize users. The geometric distribution enables us to compute the number of attempts until the first success. For example, with an advantage of 0.0199, the expected number of attempts until we observe a distinguishing event is ≈ 50. With an advantage of 0.19, the expected number of attempts the adversary needs to observe, is ≈ 5. Hence, we argue that our results are valid for both, a global as well as partially global passive adversary. In both cases, an adversary can achieve a non-negligible advantage in the Sender Receiver Pair Unlinkability game.

We conclude that the notion of (SR)$\overline{\text{L}}$ (Sender Receiver Pair Unlinkability) is not achievable for any fixed-loop ACN that utilizes loop cover traffic, fixed cascades, and sends no additional cover traffic. We can apply the hierarchy results from Kuhn et al. and conclude that it also does not achieve the Sender Message Unlinkability, Receiver Message Unlinkability (and Unobservability), and Both Side Unlinkability (and Unobservability) notions.

7 Real World Impact

We proved that a global passive adversary can break the privacy notion Sender Receiver Pair Unlinkability for any ACN that meets our three main assumptions. Additionally, we have reasoned that partially global adversaries can gain a non-negligible advantage. In this section, we evaluate the real-world impact of this finding. Firstly, we scrutinize the assumptions that led to the impossibility result. Secondly, we examine the impact on both existing and future ACNs. Finally, we will discuss potential strategies to mitigate the attack. We investigate the effects of adjusting the ACN parameters and consider weakening the three main assumptions: fixed cascades, loop cover traffic, and cover traffic.

7.1 Assumptions

The impossibility result is based on three main assumptions (A_1, A_2, and A_3) and two additional assumptions (A_4 and A_5). In the following, we demonstrate that a uniform cascade selection (A_4) is not strictly necessary for our result. Furthermore, we argue that any ACN should ensure anonymity even if two users communicate exclusively with each other (A_5). Finally, for the sake of completeness, we want to clarify that the Last Hop Attack is only applicable to ACNs that choose more than one Last Hop. We consider this to be negligible since any scalable ACN chooses more than one node to send messages to users. We therefore argue that our impossibility result mainly yields from the assumptions A_1, A_2, and A_3.

7.1.1 Main Assumptions (A_1–A_3). Firstly, note that our main assumptions: A_1: fixed cascades for the duration of an epoch [17], A_2: loop cover traffic [2, 11, 15, 32, 33], and A_3: no additional cover traffic [17], are common characteristics of ACNs. Furthermore, combining these assumptions has also been proposed [28, 35]. These assumptions are abstract and do not specify the details of the ACN, making them widely applicable. We define the epoch but do not assume anything about its duration or the number of epochs. Additionally, we do not assume anything about the length of the cascade or the type of mixing applied. We only use the fact that a message is sent from a specific mix; thereby, it holds for pool, time, stop-and-go, and even ideal mixing.

7.1.2 Cascade Selection (A_4). In order to calculate the adversary's advantage, we need to assume a form of cascade selection. So far,

we assumed a uniform cascade selection (A_4). We now show that no *scalable* cascade selection can achieve a lower advantage than the uniform cascade selection. This allows us to conclude that A_4 is not a necessary assumption for our impossibility result. We therefore introduce a definition of scalability, which depends on the maximum fraction of users that can choose the same mix as Last Hop. Based on this definition, we show that the uniform cascade selection achieves the minimal advantage for every given scalability and is, thereby, optimal. This allows us to conclude that all ACNs ensuring a given scalability have at least the same advantage as the uniform cascade selection against a global passive adversary.

We define *scalability* $S(x)$, where x can be chosen depending on the scalability and anonymity requirements of the ACN, as follows:

$S(x)$: A fraction of at most $\frac{1}{x}$ users chose the same mix as their Last Hop

We already know from Section 5.1 that the cascade selection can prevent distinguishing events, which happens if the challenge users choose the same Last Hop. For $x = 1$, a trivial and optimal strategy against the Last Hop Attack is to ensure that all users pick the same Last Hop. Thus, the adversary is unable to observe any distinguishing event, which leaves them with an advantage of zero.

For $x = 2$, every mix can, at most, serve for half of the users as Last Hop. Note that the best strategy in this setting is two Last Hops where each serves as the Last Hop for half of the users. Using more Last Hops would decrease the chances that users choose the same Last Hop and thereby increase the adversary's advantage.

We can generalize this observation: In order to minimize the advantage, we want to maximize the number of users that choose the same Last Hop. We can do this by ensuring that each Last Hop serves the maximum number of users they are allowed to, i. e., $S(x) = \frac{1}{x}$. A uniform cascade selection over x Last Hops does exactly this; it ensures that $\frac{1}{x}$ users choose the same Last Hop. We can thereby conclude that a uniform cascade selection with x Last Hops is optimal in regard to scalability.

In Figure 11, we show the advantage of a global passive adversary for a uniform cascade selection in relation to varying values of x. We can see that the attacker can achieve an advantage of 0.375 for $x = 2$. The result shows that even in a network where each mix is allowed to serve as Last Hop for half of the user base, the advantage is already non-negligible against a global passive adversary. When scaling further, e. g., $x = 10$, where each mix is allowed to serve at most a tenth of the users as Last Hop, the adversary reaches an advantage of ≈ 0.89.

We conclude that every *scalable* cascade selection can, at best, achieve the same advantage against a global passive adversary as the uniform cascade selection. This allows us to conclude that A_4 is not a necessary assumption for our impossibility result.

7.1.3 Restricting the Communication of the Challenge Users (A_5). In Section 3.4 , we restricted the capabilities of the adversary in order to analyze a realistic setting. We allowed all users to send arbitrary messages to each other and restricted only the communication of the four challenge users. For these specific users, the adversary dictates the messages they send and to whom. It is crucial to emphasize that the adversary only controls the content of these messages; the users continue to operate according to the ACN. Consequently, the adversary cannot influence the user behavior, e.g., the choice of

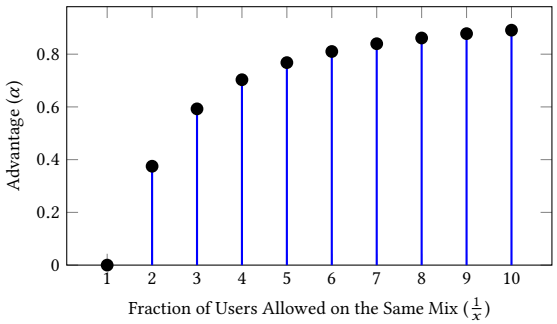

Figure 11: Advantage of a global passive adversary depending on the maximum fraction of users, a single mix can serve as Last Hop.

mixes, the timing of message transmissions, or the volume of loop cover traffic.

We argue that this is a valid and realistic assumption. An ACN should ensure anonymity for all users, irrespective of the messages exchanged or the number of communication partners involved. Therefore, the setting in which two users communicate exclusively with each other represents a realistic worst-case situation where anonymity should still be preserved.

Finally, note that without this restriction, it would be trivial to achieve strong anonymity. Allowing unrestricted communication between users enables each user to send a message to all other users in every round, achieving ultimate anonymity [22], which may not accurately reflect the anonymity an ACN provides in reality.

7.2 Consequences for ACN Designs

7.2.1 Miranda and SMRT. Since SMRT [35] is based on Miranda [28], their threat models are nearly identical. Both aim to achieve strong anonymity against a powerful adversary and assume a global observer eavesdropping on all network traffic and knowing the user's sending rates. While Miranda assumes that the adversary is, in addition, able to corrupt mixes as long as the majority of mixes are honest, SMRT assumes that the adversary is either globally passive or able to corrupt mixes. Additionally, there are slight differences in the definitions regarding the number of corrupted clients. In SMRT, the majority of clients need to be honest, while in Miranda, an arbitrary number of clients can be malicious, as long as there are 2ω honest clients, where ω is enough to ensure that any first-mix in a cascade receives a "sufficient" number of messages to ensure reasonable anonymity. Both ACNs assume the adversary can drop and delay packets on corrupted mixes, with the exclusion that they cannot drop packets between honest parties and can delay them only for a limited period. Both try to hide the correspondence between senders and receivers of a message. Additionally, Miranda specifies that they aim to provide the same protection as an "ideal mix", i.e., a single mix node that is known to be honest. Both the SMRT and Miranda satisfy all assumptions of the Last Hop Attack. They employ fixed cascades and loop cover traffic to detect if an active adversary is targeting their messages. Using loop cover messages allows them to precisely determine when each message should return. Moreover, by routing all messages over the same cascade, they ensure that if

an adversary delays a message, there is a chance that this message is a loop cover message and the attack is noticed. This approach fulfills the three main assumptions of the Last Hop Attack. They use the same cascade (A_1) for loop cover messages messages (A_2) and real messages. Additionally, they do not specify any additional cover traffic (A_3). This allows us to conclude that SMRT and Miranda are vulnerable to the Last Hop Attack. Their main design decisions focus on mitigating active attacks. Therefore, it is fair to assume that they did not describe the protocols in detail to prevent passive attacks. Nevertheless, the use of loop cover messages and fixed cascades in their design implies that real-world deployments of these networks require additional cover traffic.

7.2.2 New Mix Net Designs. The same argument can be made when designing new mix nets. Since the choice of fixed cascades and loop cover traffic is usually a very conscious decision, we think that a useful implication is that when an ACN utilizes fixed cascades and loop cover traffic, it is also required to use additional cover traffic.

Tor uses fixed cascades for its circuits. When we consider the use of loop cover traffic in Tor, for example, to detect active attacks, we can directly apply the result and conclude that this is only viable if we also deploy additional cover traffic. Note, that this impossibility result applies only when loop cover messages and the real messages are sent over the *same* fixed cascade. For example, LoopTor [33] sends loop cover traffic, but over a separate cascade, in order to obtain unobservability. Since this design uses two different cascades, it is not vulnerable to the Last Hop Attack.

7.3 Counter Measures

This section describes possible ways to mitigate the Last Hop Attack. First, we consider the effect of changing the common parameters of ACNs. Afterward, we follow the natural way of escaping an impossibility result: relaxing one of the assumptions.

Note that simply changing the parameters of the ACN will not decrease the adversary's advantage. Consider the following parameters: number of users, user traffic, delay, and length of the cascade. The number of users does not affect the proposed attack; the only factor that matters is the selection of the Last Hop of the four challenge users (u_0, u_1, u_A, u_B). Therefore, even with a large user base, the chances of the adversary succeeding remain the same as long as none of them sends to the challenge users. Similarly, the amount of user traffic, the delay (time spent at each hop), and the cascade length do not impact the attack's success rate.

We argue that increasing the number of mixes has no relevant effect on the Last Hop Attack as long as the ratio of corrupted Last Hops remains unchanged. Since the attack only uses the last mix in a cascade, adding mixes that do not serve as Last Hop has no effect on the attack. However, increasing the number of Last Hops lowers the probability that users choose the same Last Hop and thereby minimally *increases* the advantage of the adversary when the ratio of corrupted nodes remains unchanged. We can see this effect in Figure 10. The blue line in the graph shows the advantage in a network with 1,000 mixes, while the triangles represent a network with 100,000 mixes. We can observe that the triangles are either on the line or only slightly above it. This allows us to conclude that simply increasing the number of mixes does not mitigate the Last Hop Attack, as long as the share of corrupted nodes stays the same.

Relaxing one of the assumptions is a natural way of approaching an impossibility result. Since we assume loop cover messages, and fixed cascades and that there is no additional cover traffic, eliminating one of these constraints will mitigate the attack.

7.3.1 Loop Cover Traffic. The usage of loop cover traffic has proven a valuable tool in the design of mix nets [2, 11, 15, 32]. We assume that most system designers who deliberately included loop cover traffic are unwilling to sacrifice it to prevent the Last Hop Attack. Therefore we focus on relaxing the remaining assumptions.

7.3.2 Fixed Cascades. There is an ongoing discussion about the layout of mix nets [7, 16, 18], and whether fixed cascades offer an advantage. While this attack might not be strong enough to decide it definitely, it might be another argument against the usage of fixed cascades, at least in combination with loop cover traffic.

7.3.3 Drop Cover Traffic. As loop cover traffic and fixed cascades are usually deliberate design choices, we focus on the effect of additional cover traffic on the Last Hop Attack. Rather than determining the amount of cover traffic required to render the advantage negligible, we focus on the demands this cover traffic must meet. We expect that even a basic form of drop cover traffic, where users send fake messages to random other users that are dropped at reception and indistinguishable from real messages, is an efficient countermeasure against the Last Hop Attack.

This indistinguishability is both essential and difficult to achieve. If the adversary can tell the difference between loop cover traffic and drop cover traffic, they can still use the loop cover traffic to identify the Last Hops of u_0 and u_1 and then only lose if both Last Hops send a message to both challenge message receivers u_A and u_B. An analogous argument can be made if the adversary can distinguish the real traffic from the drop cover traffic. An adversary might be able to make such a distinction if the amounts of loop cover, drop cover, and real traffic differ too much. However, sending (roughly) equal amounts of cover and real traffic induces high overhead. Especially since the amount of cover traffic needed to protect all users must be based on the user who sends the most messages. Additionally, anonymity now depends on the cover traffic sent by other users. The set of Last Hops an adversary considers for a particular user u depends on the number of cover messages other users send to u over other Last Hops. Only if the amount of cover traffic is indistinguishable from the real amount, this Last Hop will be included in the set of possible Last Hops, thereby increasing the anonymity of the users.

This is very abstract since we did not specify a mechanism. Therefore, we illustrate this general problem in a more specific example without claiming generality. A user decides that, for their purpose, an amount of $\mathcal{L}$ loop messages suffices. If this value is public knowledge, we can conclude that it is known by the adversary. It might be challenging for the other users to ensure that all Last Hops send exactly $\mathcal{L}$ messages to u. But even if they are able to do this, they still do not know which Last Hop was chosen by u. So, the adversary can still identify the Last Hop, as it is the only one sending $2 \cdot \mathcal{L}$ messages to u. There are techniques to hamper this correlation, for example, by not specifying a specific amount of loop messages but a range. Also, the approach of sending according to a memoryless distribution as Loopix [32] reduces the correlation opportunities of

the adversary, but note that this problem still exists as long as the anonymity of a user relies on the cover traffic sent by other users.

Note that we considered up until this point only drop cover traffic, we did not consider other kinds of cover traffic that might be more effective. For example, the *provider* mechanism utilized in Loopix [32]. The queries and responses can be considered as cover traffic. A complete study of the effectiveness of different types of cover traffic would exceed the scope of this paper and will remain as future work.

8 Related Work

There is a body of research in the field of anonymous communication who focuses on exploring the theoretical limits of ACNs. Over the last few decades, several significant discoveries in this area have influenced our understanding of ACNs. The use of indistinguishability frameworks to establish boundaries and impossibility results has become increasingly popular. The first framework based on indistinguishability was introduced by Hevia and Micciancio [24] which defines multiple anonymity notions and relationships between them. They also present general techniques to transform protocols that achieve *weak* notions into ones that achieve stronger notions by using cover traffic. Additionally, they demonstrate that this approach is optimal in terms of message traffic.

Gelernter and Herzberg [22] proposed a framework that determines the level of anonymity achievable in a practical scenario using a top-down approach. They define a strong notion of anonymity, the *ultimate anonymity*, which requires sender anonymity and unobservability against a global passive adversary that also controls a number of corrupted participants, including destinations. To achieve this level of anonymity, they propose a protocol that has a high overhead. However, they justify this overhead by proving that any protocol that achieves ultimate anonymity also has a high overhead. They also discuss possible relaxations that require less overhead and analyze them.

One well-known bound in anonymous communication is the Anonymity Trilemma [13, 14]. It is based on the AnoA framework [4], which defines its notions based on indistinguishability games. The trilemma considers three primary objectives in anonymous communication: strong anonymity, low latency, and low bandwidth. The authors prove that it is only possible to achieve two of these goals at the same time for the most practical ACNs. The authors begin by outlining an adversary strategy. Using this strategy as a basis, they then derive an invariant that must be satisfied by any ACN looking to provide anonymity against this adversary. They then proceed to define an ideal protocol that is most effective in terms of this invariant. Finally, the authors calculate the probability of the adversary's success against their ideal protocol and, based on this, establish the limits for bandwidth and latency.

The mentioned results are noteworthy as they allow us to validate how close the potential mix net designs are to the theoretical boundaries. However, it might be challenging to apply these results when devising strategies to enhance ACNs. The Last Hop Attack is comparatively less general as it concentrates on specific characteristics of an ACN, such as loop cover traffic, cover traffic, and fixed cascades. This, in turn, makes it easier to apply the outcome to the real world. It allows us, for example, to conclude that ACNs that utilize loop cover traffic and fixed cascades need to deploy additional cover traffic in order to achieve strong privacy notions.

It is worth noting that there exist several research works that delve into specific aspects of ACNs. Oya et al. [31] conducted a study on the limits of cover traffic with respect to anonymity. There are also several papers that analyze the anonymity of onion routing [1, 5, 6, 20, 21, 27, 29], or stop-and-go-mixes [12] under certain assumptions. While these analyses ensure trust in a particular ACN, comparing them is challenging, and it is even more difficult to use the results when creating new ACNs.

Our impossibility result is due to its general ACN definition applicable to a large number of ACNs, yet still general enough to be applied to the next generations of ACNs.

To date, existing literature has primarily highlighted the positive aspects of loop cover traffic, as demonstrated in studies such as [11, 15, 32], to our knowledge, it is the first attack targeting loop cover traffic, and thereby a first step in identifying potential limitations or vulnerabilities associated with this feature.

9 Conclusion

We introduced the Last Hop Attack, which is a novel attack vector against the anonymity of fixed-loop ACNs. We observed that when an ACN sends loop cover traffic over fixed cascades, the Last Hop in each cascade sends a message to the source and destination. Based on this observation, we identified distinguishing events and presented an attack algorithm. We calculated the success probability of this algorithm and discovered that the advantage quickly becomes significant, even for partially global passive adversaries. Therefore, we concluded that the privacy notion of Sender Receiver Pair Unlinkability is impossible to achieve for any ACN that fulfills our three main assumptions: loop cover traffic, fixed cascades, and no additional cover traffic is used. We used the hierarchical results from Kuhn et al. to apply our impossibility result and further conclude that Sender Message Unlinkability, Receiver Message Unlinkability (and Unobservability), and Both Side Unlinkability (and Unobservability) are also not achievable. On the positive side of an impossibility result, its system model immediately suggests mitigation strategies. We accordingly discussed ways to mitigate the Last Hop Attack in general and the difficulties of implementing cover traffic in particular. Finally, we assert that our findings are relevant in practice by applying them to the mix net designs SMRT and Miranda. We concluded that these mix nets have to deploy cover traffic in order to provide strong anonymity.

Acknowledgments

This work has been funded by the German Research Foundation (DFG, FOR 5495, TS 477/3-1).

The authors used Grammarly[6] to revise the text in Sections 1-9 to correct typos, grammatical errors, and phrasing.

References

[1] Megumi Ando, Anna Lysyanskaya, and Eli Upfal. 2021. Practical and Provably Secure Onion Routing. arXiv:1706.05367 [cs.CR]

[2] Yawning Angel, George Danezis, Claudia Diaz, Ania Piotrowska, and David Stainton. 2017. Katzenpost Mix Network Specification. https://github.com/Katzenpost/docs/blob/master/specs/mixnet.rst

[6] www.grammarly.com

[3] Yawning Angel, George Danezis, Claudia Diaz, Ania Piotrowska, and David Stainton. 2017. *Sphinx Mix Network Cryptographic Packet Format Specification.* -. https://github.com/katzenpost/docs/blob/master/specs/sphinx.rst

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

[5] Michael Backes, Aniket Kate, Sebastian Meiser, and Esfandiar Mohammadi. 2014. (Nothing else) MATor(s): Monitoring the Anonymity of Tor's Path Selection. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 513–524. https://doi.org/10.1145/2660267.2660371

[6] Michael Backes, Sebastian Meiser, and Marcin Slowik. 2016. Your choice mator (s). *Proceedings on Privacy Enhancing Technologies* 2016, 2 (2016), 40–60.

[7] Iness Ben Guirat, Devashish Gosain, and Claudia Diaz. 2021. MiXiM: Mixnet Design Decisions and Empirical Evaluation. In *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society* (Virtual Event, Republic of Korea) *(WPES '21)*. Association for Computing Machinery, New York, NY, USA, 33–37. https://doi.org/10.1145/3463676.3485613

[8] David L Chaum. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM* 24, 2 (1981), 84–90.

[9] 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.

[10] G. Danezis, R. Dingledine, and N. Mathewson. 2003. Mixminion: design of a type III anonymous remailer protocol. In *2003 Symposium on Security and Privacy, 2003.* IEEE, IEEE, Berkeley, CA, 2–15. https://doi.org/10.1109/SECPRI.2003.1199323

[11] George Danezis and Len Sassaman. 2003. Heartbeat Traffic to Counter (n-1) Attacks: Red-Green-Black Mixes. In *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society* (Washington, DC) *(WPES '03)*. Association for Computing Machinery, New York, NY, USA, 89–93. https://doi.org/10.1145/1005140.1005154

[12] Debajyoti Das, Claudia Diaz, Aggelos Kiayias, and Thomas Zacharias. 2023. Are continuous stop-and-go mixnets provably secure? Cryptology ePrint Archive, Paper 2023/1311. https://eprint.iacr.org/2023/1311

[13] Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. 2018. Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency-choose two. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, IEEE, San Francisco, CA, USA, 108–126.

[14] Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. 2020. Comprehensive anonymity trilemma: User coordination is not enough. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (2020), 356—-383.

[15] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. 2021. The Nym Network.

[16] Claudia Diaz, Steven J. Murdoch, and Carmela Troncoso. 2010. Impact of Network Topology on Anonymity and Overhead in Low-Latency Anonymity Networks. In *Privacy Enhancing Technologies*, Mikhail J. Atallah and Nicholas J. Hopper (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 184–201.

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

[18] Roger Dingledine, Vitaly Shmatikov, and Paul Syverson. 2005. Synchronous Batching: From Cascades to Free Routes. In *Privacy Enhancing Technologies*, David Martin and Andrei Serjantov (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 186–206.

[19] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–12.

[20] Joan Feigenbaum, Aaron Johnson, and Paul Syverson. 2007. A Model of Onion Routing with Provable Anonymity. In *Financial Cryptography and Data Security*, Sven Dietrich and Rachna Dhamija (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 57–71.

[21] Joan Feigenbaum, Aaron Johnson, and Paul Syverson. 2012. Probabilistic analysis of onion routing in a black-box model. *ACM Transactions on Information and System Security (TISSEC)* 15, 3 (2012), 1–28.

[22] Nethanel Gelernter and Amir Herzberg. 2013. On the limits of provable anonymity. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society* (Berlin, Germany) *(WPES '13)*. Association for Computing Machinery, New York, NY, USA, 225–236. https://doi.org/10.1145/2517840.2517850

[23] Iness Ben Guirat, Claudia Diaz, Karim Eldefrawy, and Hadas Zeilberger. 2024. Traffic Analysis by Adversaries with Partial Visibility. In *Computer Security – ESORICS 2023*, Gene Tsudik, Mauro Conti, Kaitai Liang, and Georgios Smaragdakis (Eds.). Springer Nature Switzerland, Cham, 338–358.

[24] Alejandro Hevia and Daniele Micciancio. 2008. An Indistinguishability-Based Characterization of Anonymous Channels. In *Privacy Enhancing Technologies*, Nikita Borisov and Ian Goldberg (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 24–43.

[25] 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.

[26] Christiane Kuhn, Martin Beck, Stefan Schiffner, Eduard Jorswieck, and Thorsten Strufe. 2019. On Privacy Notions in Anonymous Communication. *Proceedings on Privacy Enhancing Technologies* 2019 (04 2019), 105–125. https://doi.org/10.2478/popets-2019-0022

[27] Christiane Kuhn, Martin Beck, and Thorsten Strufe. 2020. Breaking and (Partially) Fixing Provably Secure Onion Routing. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA, 168–185. https://doi.org/10.1109/SP40000.2020.00039

[28] Hemi Leibowitz, Ania M. Piotrowska, George Danezis, and Amir Herzberg. 2019. No Right to Remain Silent: Isolating Malicious Mixes. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 1841–1858. https://www.usenix.org/conference/usenixsecurity19/presentation/leibowitz

[29] Alessandro Melloni, Martijn Stam, and Øyvind Ytrehus. 2021. On Evaluating Anonymity of Onion Routing. In *Selected Areas in Cryptography: 28th International Conference, Virtual Event, September 29 – October 1, 2021, Revised Selected Papers*. Springer-Verlag, Berlin, Heidelberg, 3–24. https://doi.org/10.1007/978-3-030-99277-4_1

[30] Ulf Moller, Lance Cottrell, Peter Palfrader, and Len Sassaman. 2004. Mixmaster: anonymous remailer. https://mixmaster.sourceforge.net/. Accessed: 2023-11-21.

[31] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. 2014. Do Dummies Pay Off? Limits of Dummy Traffic Protection in Anonymous Communications. In *Privacy Enhancing Technologies*, Emiliano De Cristofaro and Steven J. Murdoch (Eds.). Springer International Publishing, Cham, 204–223.

[32] 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)*. USENIX Association, Vancouver, BC, 1199–1216. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/piotrowska

[33] Jaume Planas Planas. 2020. *LOOPTOR: Fighting traffic correlation on the Tor network.* B.S. thesis. Universitat Politècnica de Catalunya.

[34] John A Rice. 2006. *Mathematical statistics and data analysis*. Duxbury Press, Belmont, Calif.

[35] Jincai Zou, Zhihan Tan, Yaya Huang, Yixing Chen, Yuqiang Zhang, and Ning Hu. 2022. SMRT:An Effective Malicious Node Resistance Design for Mixnets. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*. IEEE, Berlin, Heidelberg, 110–117. https://doi.org/10.1109/DSC55868.2022.00022

A Artifacts and Code

In order to ensure easy reproducibility of our results, we provide the code that we created for this paper. You can find the calculations we used for Figures 2, 9, 10, 11, 12, and 13 on GitHub: https://github.com/Ti-ger/Last_Hop_Attack.

As well as our WolframAlpha queries in the following.
WolframAlpha query to calculate the adversary's advantage:

```
N =1000;c = 1;
(1 - binomial(N-2 ,c)/binomial(N ,c))
* (1 - 1/N)^3
+ (1 - binomial(N-1,c)/binomial(N ,c))
* 2 * (1 - 1/N)^2 * 1/N
```

WolframAlpha query to calculate the adversary's advantage in the bandwidth analysis:

```
x = 0.02, k = 10; 2 * x - 4 * x^2 * 1/k
+ 2 *x^3 *(1/k)^2 - x*(x *(k-1)/k)
* (1 - x*(1/k)) * (1 - (x * (1/k)))
```

B Differential Private Bounds

The framework of Kuhn et al. [26] enables the definition of bounds in the form of differential privacy.

$$\Pr(g = 0|b = 0) \leq e^{\varepsilon} \cdot \Pr(g = 0|b = 1) + \delta$$

Here, we can insert the previous (Section 5.5) results.

$$\Pr(\mathcal{D}_p) \cdot 1 + (1 - \Pr(\mathcal{D}_p)) \cdot \frac{1}{2} \leq e^{\varepsilon} \cdot (1 - \Pr(\mathcal{D}_p)) \cdot \frac{1}{2} + \delta$$

In order to calculate bounds for fixed delta values, for example, $\delta = 0$, this can be transformed to:

$$\delta \geq \Pr(\mathcal{D}_p)\left(\frac{e^{\varepsilon}}{2} + \frac{1}{2}\right) - \frac{e^{\varepsilon}}{2} + \frac{1}{2}$$

With these equations, we can calculate the achievable bounds in the metric of differential privacy for a given ACN.

C Bandwidth-based Cascade Selection

In our paper, we considered a uniform selection of the mixes in the cascade (A_4) as well as a uniform selection of corrupted mixes. In the following, we sketch that the Last Hop Attack is also viable when other types of cascade selection are used. We first consider the cascade selection with bandwidth weights. Here, clients select the nodes for their cascade based on the node's bandwidth capacity. This technique is commonly used, for example, in Tor [17]. We denote the total bandwidth available in the network $\mathcal{B}_\mathrm{t}$ and the fraction of this bandwidth that is controlled by the adversary $\mathcal{B}_\mathrm{A}$, i. e., their *attack budget*. In general, the adversary can freely choose how much bandwidth each of their mixes provides. We, however, assume for simplicity that the attack budget is uniformly distributed on all corrupted nodes. We denote the number of corrupted nodes with k, Note that the optimal strategy for the adversary is to spread the bandwidth on as many nodes as possible to prevent collisions. In order to calculate realistic results, we will limit the maximum number of corrupted nodes by k.

We denote the Last Hop of a user u as L_u and the corrupted nodes by the adversary as c_x with $x \in [1, k]$. We previously argued that the only two Last Hops that can witness distinguishing events are the Last Hops of users u_0 and u_1. For the following explanation, we will focus on user u_0 in scenario $b = 0$ and then continue with the general case.

Based on our assumption, we can easily determine the probability of user u_0 choosing one of the corrupted Last Hops by calculating:

$$\Pr(L_{u_0} = C_x) = \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \quad \text{with } x \in [1, k]$$

We also argued that distinguishing events can be denied based on the cascade selection. This would be the case for user u_0 in $b = 0$ when u_0 and u_1 chose the same Last Hop or when u_0 and u_B chose the same Last Hop.

We can calculate these probabilities for $x \in [1, k]$ as follows:

$$\begin{aligned}\Pr(L_{u_0} = C_x \wedge L_{u_1} = C_x) &= \Pr(L_{u_0} = C_x \wedge L_{u_B} = C_x) \\ &= \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \cdot \frac{1}{k} \cdot \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \cdot \frac{1}{k} \cdot k = \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k}\end{aligned}$$

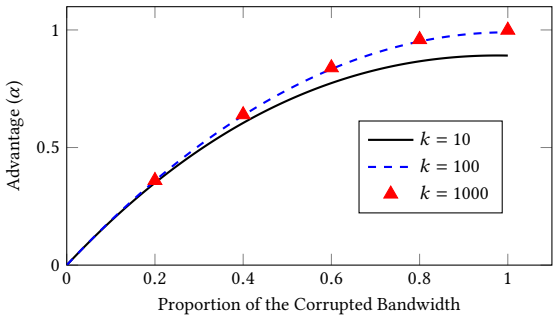

Figure 12: Advantage of the adversary depending on the proportion of the corrupted bandwidth for different k.

We can now calculate the probability of witnessing a distinguishing event on the Last Hop of user u_0 as

$$\begin{aligned}\Pr(\mathcal{D}_{L_{u_0}}) =& \Pr(L_{u_0} = C_x) \\ &- \Pr(L_{u_0} = C_x \wedge L_{u_1} = C_x) \\ &- \Pr(L_{u_0} = C_x \wedge L_{u_B} = C_x) \\ &+ \Pr(L_{u_0} = C_x \wedge L_{u_B} = C_x \wedge L_{u_0} = C_x \wedge L_{u_1} = C_x) \\ =& \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} - \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} - \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} + \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^3 \cdot \left(\frac{1}{k}\right)^2 \\ =& \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} - 2 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} + \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^3 \cdot \left(\frac{1}{k}\right)^2\end{aligned}$$

We can calculate this probability analogously for user u_1 in scenario $b = 0$:

$$\begin{aligned}\Pr(\mathcal{D}_{L_{u_1}}) =& \Pr(L_{u_1} = C_x) \\ &- \Pr(L_{u_1} = C_x \wedge L_{u_0} = C_x) \\ &- \Pr(L_{u_1} = C_x \wedge L_{u_A} = C_x) \\ &+ \Pr(L_{u_1} = C_x \wedge L_{u_A} = C_x \wedge L_{u_1} = C_x \wedge L_{u_0} = C_x) \\ =& \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} - 2 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} + \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^3 \cdot \left(\frac{1}{k}\right)^2\end{aligned}$$

This enables us to calculate the probability of a distinguishing event:

$$\begin{aligned}\Pr(\mathcal{D}) =& \Pr(\mathcal{D}_{L_{u_0}}) + \Pr(\mathcal{D}_{L_{u_1}}) - \Pr(\mathcal{D}_{L_{u_0}} \wedge \mathcal{D}_{L_{u_0}}) \\ =& \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} - 2 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} + \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^3 \cdot \left(\frac{1}{k}\right)^2 \\ &+ \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} - 2 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} + \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^3 \cdot \left(\frac{1}{k}\right)^2 \\ &- \Pr(L_{u_0} = C_x) \cdot \Pr(L_{u_1} = C_y, x \neq y) \cdot \Pr(L_{u_B} \neq C_x) \cdot \Pr(L_{u_A} \neq C_y) \\ =& 2 \cdot \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} - 4 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} + 2 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^3 \cdot \left(\frac{1}{k}\right)^2 \\ &- \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \cdot \frac{k-1}{k}\right) \cdot \left(1 - \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \cdot \frac{1}{k}\right) \cdot \left(1 - \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \cdot \frac{1}{k}\right) \\ =& 2 \cdot \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} - 4 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{1}{k} + 2 \cdot \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^3 \cdot \left(\frac{1}{k}\right)^2 \\ &- \left(\frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}}\right)^2 \cdot \frac{k-1}{k} \cdot \left(1 - \frac{\mathcal{B}_\mathrm{A}}{\mathcal{B}_\mathrm{t}} \cdot \frac{1}{k}\right)^2\end{aligned}$$

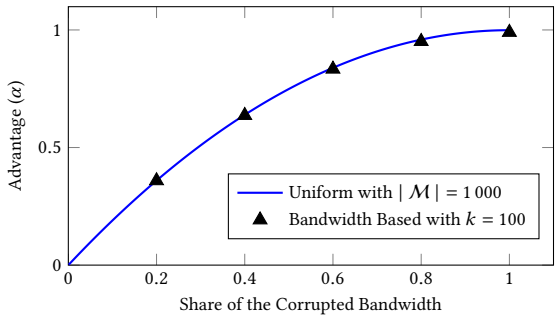

Figure 13: Advantage of the adversary for uniform and bandwidth-based cascade selection.

In the graph shown in Figure 12, the advantage of an adversary (y-axis) is plotted against the proportion of corrupted nodes (x-axis) for different values of k (10, 100, and 1,000). The specific queries used for this analysis can be found in Appendix A.

The graph shows that if an adversary controls 20 percent of the total bandwidth distributed over ten mixes, they already have an advantage of ≈ 0.35. For a global passive adversary with 100 mixes, the advantage is close to one (≈ 0.9899) and for 1,000 mixes, the advantage is even higher (≈ 0.9989).

Using the presented formula, we can calculate the advantage of an adversary who can only observe ten mixes with a total bandwidth of one percent as ≈ 0.0199.

We already introduced formulas for the uniform cascade selection in Section 5.3 In Figure 13, we compare the advantages of both mechanisms.

The x-axis depicts the amount of bandwidth observable by the adversary, and the y-axis their advantage. The (blue) solid line indicates the advantage when considering a uniform cascade selection in a network with 1,000 nodes in total. The (black) triangles depict the advantage for a bandwidth-based cascade selection, where the adversary can observe 100 nodes. We can see that both techniques are vulnerable once the adversary is able to observe a sufficient proportion of the network.